Metody numeryczne

Jacek Kobus

Wydział Fizyki, Astronomii i Informatyki Stosowanej UMK (2013/2014)

http://www.fizyka.umk.pl/~jkob/mn[4].pdf

Czym jest analiza numeryczna?

- Przedmiotem analizy numerycznej (metod numerycznych, *scientific computing*) jest tworzenie, projektowanie i analiza algorytmów służących rozwiązywaniu problemów matematycznych pojawiających się w zagadnieniach naukowych i inżynieryjnych.
- Zagadnienia te opisane są funkcjami i równaniami, które zależą od ciągłych z natury wielkości (położenie, prędkość, temperatura, itp.).
- Zazwyczaj zagadnienia te nie mają ścisłych, analitycznych rozwiązań. Dlatego potrzebne są metody rozwiązywania, które pozwalają na znalezienie rozwiązania (przybliżonego) w skończonej liczbie kroków.
- Kluczowy problem: analiza przybliżeń i ich efektów.
- Poszukiwanie stabilnych i efektywych algorytmów.

Strategia postępowania: zastępowanie trudnego problemu przez prostszy, który ma identyczne, bądź zbliżone rozwiązanie

Zastępowanie:

- procesu nieskończonego przez skończony
- ogólnych macierzy przez prostsze
- skomplikowanych funkcji przez prostsze (np. wielomiany)
- zagadnień nieliniowych przez liniowe
- zastępowanie równań różniczkowych przez algebraiczne
- zagadnień nieskończenie wymiarowych przez zagadnienia określone w przestrzeniach skończenie wymiarowych

Metody numeryczne

1. Wstęp – rola metod numerycznych w rozwiązywaniu zadań
2. Dokładność obliczeń numerycznych
3. Układy równań liniowych
4. Aproksymacja liniowa sredniokwadratowa
5. Wartości i wektory własne macierzy
6. Równania nieliniowe
7. Interpolacja
8. Różniczkowanie numeryczne
9. Całkowanie numeryczne
10. Równania różniczkowe zwyczajne
11. Równania różniczkowe cząstkowe

Literatura

- J. Stoer, R. Bulirsch: *Wstęp do analizy numerycznej*, PWN 1987
- G. Dahlquist, A. Björck: *Metody numeryczne*
- A. Ralston: *Wstęp do analizy numerycznej*
- M. T. Heath, *Scientific Computing*, McGraw-Hill, 1997
- R. L. Burden, J. D. Faires: *Numerical Analysis*
- D. M. Young, R. T. Gregory: *A survey of numerical mathematics*
- W. H. Press i in.: *Numerical Recipes (the Art of Scientific Computing)*
- A. Kiełbasiński, H. Schwetlick: *Numeryczna algebra liniowa*, WNT 1992
- D. Kincaid, W. Cheney: *Analiza numeryczna*, WNT 2006
- Z. Fortuna, B. Macukow, J. Wąsowski: *Metody Numeryczne*, WNT 1993

Źródła błędów

1. **Modelowanie:** upraszczające założenia tkwiące u podstaw modelu matematycznego użytego do opisu danego zjawiska
2. **Dane doświadczalne:** niedokładności danych wejściowych wynikające z ograniczeń urządzeń pomiarowych i przypadkowych zmian warunków, w których dokonywane są pomiary.
3. **Poprzednie obliczenia:**
   - błędy zaokrągleń
   - błędy przybliżeń (aproksymacji, dyskretyzacji, obcięcia)

$$e^x \approx 1 + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!}$$

$$f(x) \approx f(x_0) + \frac{f(x) - f(x_0)}{x - x_0}(x - x_0)$$

Błędy w obliczeniach

Rozwiązanie problemu można przedstawić (w przybliżeniu) jako

$$f : \mathbb{R} \to \mathbb{R}$$

Dla dokładnej danej wejściowej x otrzymujemy prawidziwy wynik $f(x)$. Jeśli dysponujemy tylko $\tilde{x} \approx x$ oraz możemy posłużyć się przybliżonym rozwiązaniem $\hat{f}$, to

$$\begin{aligned} \text{całkowity błąd} &= \hat{f}(\tilde{x}) - f(x) \\ &= \underbrace{(\hat{f}(\tilde{x}) - f(\tilde{x}))}_{(a)} + \underbrace{(f(\tilde{x}) - f(x))}_{(b)} \end{aligned}$$

(a) błąd obliczeniowy (*computational errror*)

(b) propagowany błąd danych (*propagated data error*)

Rodzaje błędów

- Błąd obcięcia/dyskretyzacji: różnica między wynikiem prawdziwym i uzyskanym przy pomocy danego algorytmu z zastosowaniem dokładnej arytmetyki
- Błąd zaokrąglenia: różnica między wynikiem uzyskanym przy pomocy danego algorytmu z zastosowaniem arytmetyki dokładnej i arytmetyki o skończonej dokładności

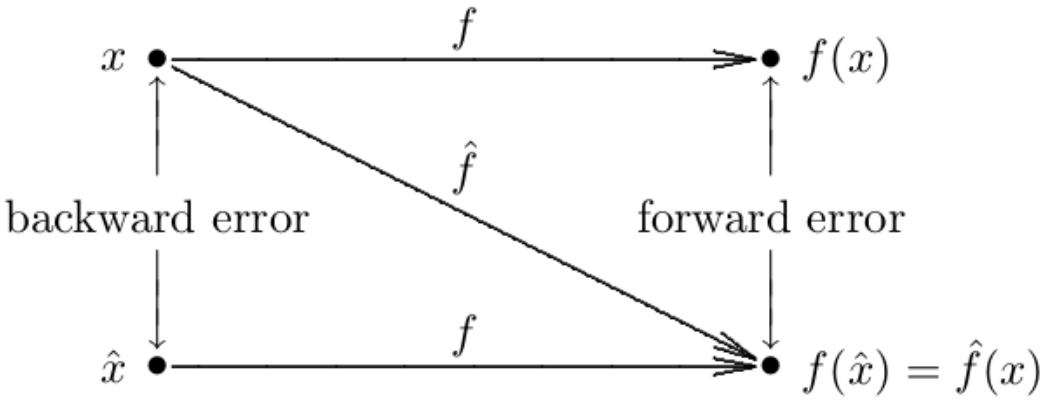
$$\text{błąd obliczeniowy} = \text{błąd obcięcia} + \text{błąd zaokrąglenia}$$

W praktyce zwykle trudno wydzielić poszczególne typy błędów i dlatego często traktuje sie je zbiorczo jako błędy danych wejściowych.

Wsteczna analiza błędów[1]

Analiza błędów obliczeniowych jest zwykle bardzo złożna. Dlatego przeprowadza się wsteczną analizę błędów.

Załóżmy, że rozwiązanie które mamy jest ścisłym rozwiązanie dla zmodyfikowanego problemu. O ile trzeba zmodyfikować dane wejściowe, żeby otrzymać taki sam wynik dla problemu oryginalnego?



[1] M. T. Heath, *Scientific Computing*, McGraw-Hill, 1997

Rodzaje błędów

Niech $\tilde{a}$ będzie przybliżoną wartością wielkości, której dokładną wartość oznaczmy przez a. Określamy dwa rodzaje błędów:

- błąd bezwzględny wartości $\tilde{a}$: $\tilde{a} - a$
- błąd względny wartości $\tilde{a}$: $(\tilde{a} - a)/a$, jeśli $a \neq 0$
  wielkość bezwymiarowa, często wyrażana w procentach

W praktyce zwykle nie znamy wartości dokładnej. Dlatego zastępujemy ją wartością daną i obserwujemy jak ulega zmianie pod wpływem rozmaitych błędów.

Wskaźniki uwarunkowania dla zadań i algorytmów

- dobre/złe warunkowanie algorytmów
- dobre/złe warunkowanie zadań

Dobre/złe warunkowanie jest określone przez niewrażliwość/wrażliwość danych wyjściowych na zaburzenie danych wejściowych.

Wskaźnik uwarunkownia problemu f w punkcie x jest zdefiniowany jako

$$\frac{|(f(\tilde{x}) - f(x))/f(x)|}{|(\tilde{x} - x)/x|}$$

Mówimy o niestabilnych (numerycznie) algorytmach lub niestabilnych (matematycznie) zadaniach, jeśli liczba uwarunkowania jest dużo większa niż jeden.

Przykład: $e^{\pm x}$ dla $x \gg 1$, $\cos(x)$ dla $x \approx \pi/2$.

Wskaźniki uwarunkowania dla zadań i algorytmów

Przykład: Należy wyznaczyć x, y oraz $x+y$ z układu równań

$$\begin{aligned} x + \alpha y &= 1 \\ \alpha x + y &= 0 \end{aligned}$$

$x = 1/(1-\alpha^2)$ i wskaźnik uwarunkowania dla x wynosi

$$\frac{\Delta x/x}{\Delta\alpha/\alpha} = \frac{2\alpha^2}{1-\alpha^2}$$

Algorytm jest dobrze uwarunkowany dla $\alpha \ll 1$ i $\alpha \gg 1$, źle dla $\alpha \approx 1$.

Obliczanie $y = -\alpha/(1-\alpha^2)$ jest źle uwarunkowane dla $\alpha^2 \approx 1$.

$$z = x + y = \frac{1}{1-\alpha^2} - \frac{\alpha}{1-\alpha^2} = \frac{1}{1+\alpha}$$

jest dobrze uwarunkowane dla $\alpha \approx 1$.

Złe uwarunkowanie algorytmu można często zmniejszyć przez przeprowadzienie obliczeń w podwyższonej precyzji. Wskaźniki uwarunkowania algorytmu i zadania są od siebie wzajemnie niezależne.

Cyfry istotne i znaczące

Jak poprawnie zapisywać wyniki pomiarów/obliczeń?

- Czy masa samochodu wynosi 1.25 t, czy 1250000 g?
- Czy masa próbki chemicznej wynosi 2.3562 g, czy 0.0023562 kg?

W jaki sposób zapis wyniku pomiaru ma określać jego dokładność?

Cyfry istotne i znaczące

- Cyfry istotne – wszystkie cyfry z wyjątkiem zer na początku liczby pomagających określić pozycję kropki
- Cyfry ułamkowe – wszystkie cyfry po kropce, także zera między kropką i pierwszą cyfrą różną od zera
- Poprawne cyfry ułamkowe, cyfry znaczące

  Niech wartość wielkości a będzie wyrażona liczbą posiadającą cyfry ułamkowe. Jeśli

  $$|\tilde{a} - a| < \frac{1}{2} 10^{-p},$$

  to cyfry ułamkowe występujące aż do pozycji p-tej nazywamy poprawnymi cyframi ułamkowymi.

  Cyfry znaczące, to cyfry istotne wchodzące w skład poprawnych cyfr ułamkowych; cyfry istotne występujące w $\tilde{a}$ do pozycji $p-$jej po kropce.

Cyfry istotne i znaczące

wynik	cu	ci	pcu	cz	$\lvert\tilde{a}/a - 1\rvert$	poprawny wynik
0.0012345 ± 0.0000002	7	5	6	4	2×10^{-4}	1.234×10^{-3}
0.0012345 ± 0.00000004	7	5	7	5	3×10^{-5}	1.2345×10^{-3}
0.001230045 ± 0.00006	9	7	3	1	5×10^{-2}	1×10^{-3}
123.45 ± 0.002	2	5	2	5	2×10^{-5}	1.2345×10^{2}
123.45 ± 0.008	2	5	1	4	6×10^{-5}	1.235×10^{2}
123.4500 ± 0.002	4	7	1	4	2×10^{-5}	1.235×10^{2}
123.4005 ± 0.0002	4	7	3	6	2×10^{-6}	1.23401×10^{2}
123.4005 ± 0.002	4	7	2	4	2×10^{-5}	1.2340×10^{2}
123.45 ± 0.06	2	5	0	3	5×10^{-4}	1.23×10^{2}
0.123045 ± 0.000002	6	6	5	5	2×10^{-5}	1.2305×10^{-1}
0.123045 ± 0.0000004	6	6	6	6	3×10^{-6}	1.23045×10^{-1}
0.123045 ± 0.00006	6	6	3	3	5×10^{-3}	1.23×10^{-1}

cu – cyfry ułamkowe, ci – cyfry istotne, pcu – poprawne cu, cz – cyfry znaczące

Cyfry poprawne określają wielkość błędu bezwzględnego, cyfry znaczące – względnego.

Jeśli jakiś wynik jest zapisywany bez oszacowania błędu, ale poprawnie zapisany, np. 123.45, to mamy prawo przypuszczać, że jest on obarczony błędem nie większym niż 0.005, czyli obejmuje on liczby z zakresu od 123.445 do 123.455.

Zaokrąglanie liczb

- przybliżanie liczby niewymiernej przez wymierną, np. π przez $22/7$
- przybliżanie ułamka przez rozwinięcie dziesiętne okresowe, np. $5/3$ przez 1.6667
- redukowanie liczby cyfr istotnych ułamka
- zastępowanie końcowych cyfr liczby całkowitej przez zera, np. 23217 przez 23200
- zamiana liczb rzeczywistych na całkowite
- przeprowadzanie operacji arytmetycznych na liczbach o skończonej precyzji

$$a \times b = 0.122 \times 0.915 = 0.111630 \approx 0.112$$

$$c \times d = 0.126 \times 0.923 = 0.116298 \approx 0.116$$

Zaokrąglanie liczb

Ograniczenie liczby cyfr ułamkowych odbywa się poprzez obcięcie lub zaokrąglenie.

Obcięcie polega na odrzuceniu cyfry $p+1$-szej i następnych.

Zaokrąglenie (do najbliższej) polega na wyborze spośród liczb mających p-cyfr ułamkowych liczby najbliższej do liczby danej. Jeśli fragment liczby znajdującej się na prawo od cyfry p-tej ma moduł mniejszy niż $\frac{1}{2}\times 10^{-p}$, to cyfrę p-tą pozostawiamy bez zmiany. W przeciwnym przypadku zwiększamy tę cyfrę o 1.

W standardzie IEEE 754 wyróżnia się przypadek, kiedy moduł fragmentu jest dokładnie równy $\frac{1}{2}\times 10^{-p}$. Wówczas zwiększa się cyfrę p-tą o 1, jeśli jest nieparzysta, a pozostawia się ją bez zmiany, jeśli jest parzysta. Jest to tzw. zaokrąglanie połówki do parzystej (*rounding half to even*).

Zaokrąglanie liczb

Konsekwentne stosowanie zaokrąglania oznacza, że wyniki liczbowe nie uzupełnione dodatkowym oszacowaniem błędu należy uważać za obarczone błędem nie większym niż $\frac{1}{2}$ jednostki ostatniego uwzględnionego miejsca ułamkowego.

Przykład:
Jeśli w wyniku pomiaru otrzymujemy wartość 8, to wynik działania 8×8 należy zapisać jako 6×10^1.

Wynik działania nie może mieć więcej cyfr znaczących niż liczby wchodzące w jego skład.

Podanie wyniku jako 6.4×10^1 dawałoby fałszywe poczucie dokładności pomiaru. Jeśli pomiar daje wyniki z zakresu $[7.5, 8.5]$, to jego kwadrat obejmuje liczby z zakresu $[56.25, 72, 25]$.

Przenoszenie się błędów

Podstawowymi operacjami arytmetycznymi wykonywanymi przez maszynę cyfrową są dodawanie, odejmowanie, mnożenie, dzielenie. Z tymi operacjami są związane błędy wynikające z niedokładności składników/czynników (operandów) i błędów zaokrągleń.

$x_1/\Delta x_1$, $x_2/\Delta x_2$, $\ldots$ $x_n/\Delta x_n$ – operandy/błędy operandów

W ogólnym przypadku wyznaczona (obliczona) wielkość

$$y = f(x_1, x_2, \ldots, x_n)$$

jest pewną funkcją wielu niezależnych zmiennych x_i, z których każda jest obarczona błędem Δx_i.

Ze wzoru Taylora wynika, że

$$\begin{aligned} y + \Delta y &= f(x_1 + \Delta x_i, x_2 + \Delta x_2, \ldots, x_n + \Delta x_n) \\ &= f(x_1, x_2, \ldots, x_n) + \sum_{i=1}^{n} \frac{\partial f}{\partial x_i} \Delta x_i + \sum_{i=1}^{n} \sum_{j=1}^{n} \frac{\partial^2 f}{\partial x_i \partial x_j} \Delta x_i \Delta x_j + \ldots \end{aligned}$$

Przenoszenie się błędów

Jeśli założyć, że błędy Δx_i są małe, to

$$\Delta y \approx \sum_{i=1}^{n} \frac{\partial f}{\partial x_i} \Delta x_i$$

$$\Delta y \lesssim \sum_{i=1}^{n} \left|\frac{\partial f}{\partial x_i}\right| |\Delta x_i|$$

Jest to ogólny wzór dla szacowania błędów (przenoszenia się błędów).

Przenoszenie się błędów

Ze wzoru wynika, że dla sumy/różnicy dwóch wielkości x_1 i x_2 obarczonych błędami Δx_1 i Δx_2 otrzymujemy

$$y = x_1 \pm x_2$$

$$\Delta y = \Delta x_1 \pm \Delta x_2$$

Ponieważ błędy mogą być dodatnie bądź ujemne, więc przy szacowaniu błędu sumy musimy wziąć pod uwagę ich moduły, czyli błąd bezwzględny y wynosi

$$\Delta y \leqslant |\Delta x_1| + |\Delta x_2|$$

Przenoszenie się błędów

Błąd względny przy dodawaniu

$$\frac{\Delta y}{y} = \frac{|\Delta x_1| + |\Delta x_2|}{x_1 + x_2}$$

Błąd względny przy odejmowaniu

$$\frac{\Delta y}{y} = \frac{|\Delta x_1| + |\Delta x_2|}{x_1 - x_2}$$

Jeśli $x_1 \approx x_2$, to błąd względny może być dużo większy niż błędy względne obu składników!

Przykład: $\log \frac{a}{b} = \log a - \log b$, $a = 2.5000$ i $b = 2.4999$.

$$\log \frac{a}{b} = 0.39794 - 0.39792 = 0.00002,$$

Żadna cyfra wyniku nie jest poprawna, gdyż powinno być 0.000017372!

Problem: Odejmowanie bliskich sobie liczb, to główne źródło błędów.

Przenoszenie się błędów

Jak chronić się przed utratą cyfr znaczących?

- stosować wzory stabilne numerycznie:

$$\cos(x_1) - \cos(x_2) = 2 \sin \frac{1}{2}(x_1 + x_2) \sin \frac{1}{2}(x_1 - x_2)$$

.

$$\ln(x - \sqrt{x^2 - 1}) = \ln \frac{1}{x + \sqrt{x^2 - 1}}$$

- stosować wzór Taylora:

$$f(x + \Delta x) - f(x) = f(x + \Delta x) - f(x) = f'(x)\Delta x + f''(x)\Delta x^2 + \ldots$$

- prowadzić rachunki w podwyższonej precyzji

Błąd maksymalny i statystyczny

Dla $y = \Sigma_{i=1}^{n} x_i$ błąd wynosi $\Delta y = \Sigma_{i=1}^{n} \Delta x_i$.

Błąd sumy zależy liniowo od liczby składników. Jeśli $\Delta x_i = 0.5$, to suma 10^4 wyrazów będzie obarczona błędem 5×10^3!

W praktyce nie wszystkie błędy mają taki sam znak, bo zaokrąglanie wprowadza błędy dodatnie i ujemne, które się kompensują.

W przypadku dużej liczby zmiennych błąd wielkości złożonej szacujemy przez podanie tzw. błędu standardowego (statystycznego).

Takie postępowanie jest uzasadnione, jeśli możemy traktować błędy poszczególnych zmiennych jako zmienne losowe niezależne o rozkładzie normalnym.

Błąd maksymalny i statystyczny

Tw. Załóżmy, że błędy Δx_i, $i = 1, \ldots, n$ są niezależnymi zmiennymi losowymi o wartości oczekiwanej równej zeru i odchyleniach standardowych ε_i. Wtedy błąd standardowy dla $y = f(x_1, x_2, \ldots, x_n)$ jest dany wzorem

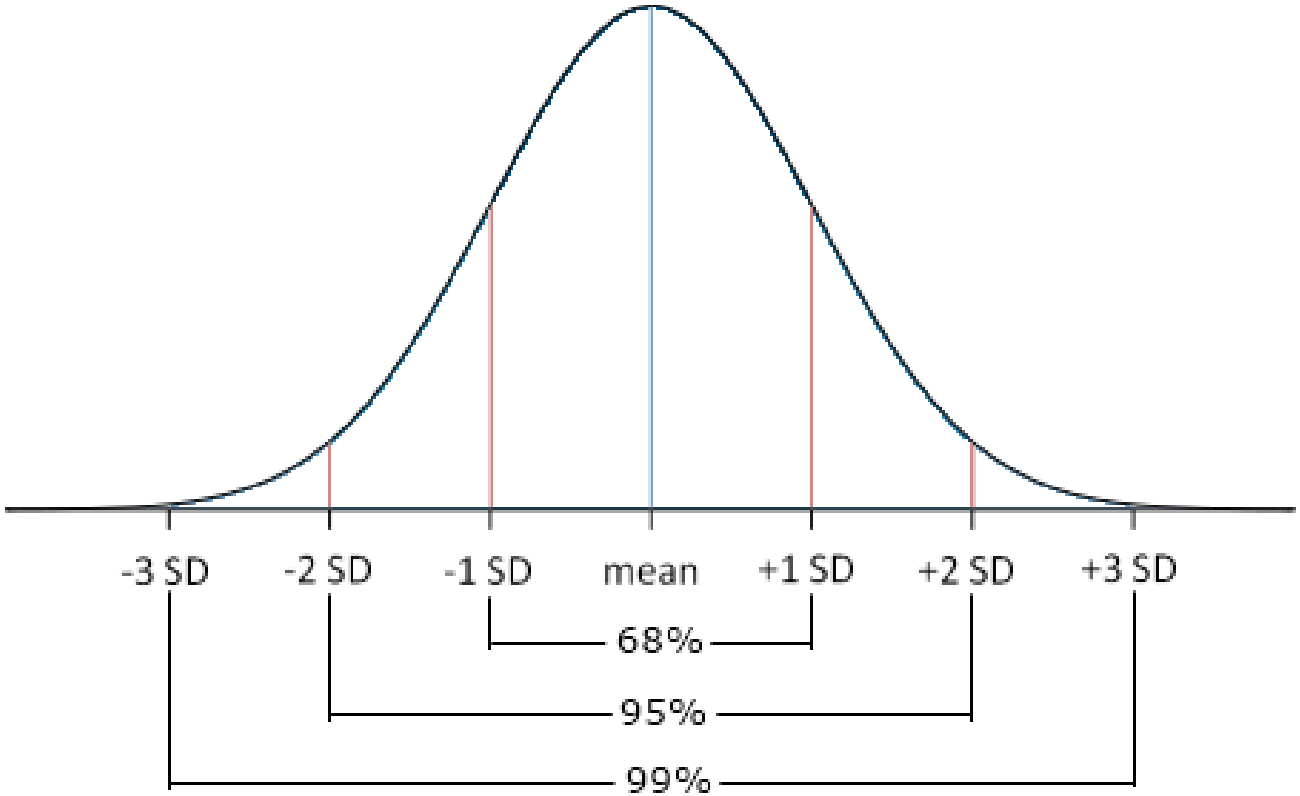
$$\left[\sum_{i=1}^{n} \left(\frac{\partial f}{\partial x_i} \right)^2 \varepsilon_i^2 \right]^{1/2}$$

Dla sumy n-składników o jednakowym błędzie ε błąd standardowy wynosi $\sqrt{\Sigma_{i=1}^{n} \varepsilon^2} = \varepsilon\sqrt{n}$.

Przy powyższych założeniach i $n \gg 1$ błąd sumy y ma w przybliżeniu rozkład normalny z wartością oczekiwaną 0 i odchyleniem standardowym $\sigma = \varepsilon\sqrt{n}$ i jest opisany wzorem

$$\rho(\Delta y) = \frac{1}{\sqrt{2\pi}\sigma} \exp(-\Delta y^2/2\sigma^2)$$

Błąd maksymalny i statystyczny[2]



[2] http://www.totallab.com/products/samespots/support/images/normal-distribution.png

Układy pozycyjne

Dowodzi się, że każda liczba rzeczywista ma jedyną reprezentację postaci

$$d_n\beta^n + d_{n-1}\beta^{n-1} + \ldots + d_1\beta^1 + d_0\beta^0 + d_1\beta^{-1} + d_2\beta^{-2} + \ldots$$

gdzie współczynniki są cyframi układu, tj. liczbami naturalnymi (całkowitymi) takimi, że $0 \leqslant d_i < \beta$; β – podstawa układu pozycyjnego.

Im niższa jest podstawa układu pozycyjnego tym prostsze są reguły wykonywania dodawania i mnożenia.

Większość komputerów pracuje w oparciu o dwójkowy układ pozycyjny.[3]

Znormalizowane reprezentacje liczby x przy podstawie β

$$x = m \times \beta^e, \quad 1/\beta \leqslant |m| < 1, \quad x \neq 0$$

$$x = m \times \beta^e, \quad 1 \leqslant |m| < \beta, \quad x \neq 0$$

[3]Liczby zapisane w układzie dwójkowym są około 3.3 razy dłuższe niż w układzie dziesiętnym. Przy pomocy p cyfr dwójkowych (bitów) można zapisać liczbę rzędu 2^p. Ponieważ $2^p = 10^q$, więc $q = p \log 2 = p\,0.30 = p/3.3$.

Liczby zmiennopozycyjne

Niech liczba rzeczywista będzie postaci

$$x = \pm \left(d_0 + d_1\beta^{-1} + d_2\beta^{-2} + \ldots + d_{p-1}\beta^{-(p-1)} \right) \beta^e$$

gdzie

$$e_{\min} \leqslant e \leqslant e_{\max}, \qquad 0 \leqslant d_i < \beta, \quad i = 0, 1, \ldots, p-1$$

- β – podstawa systemu pozycyjnego (liczba parzysta)
- e – wykładnik (cecha)
- $e_{\min}$ i $e_{\max}$ – zakres wykładnika
- $d_0 d_1 \ldots d_{t-1}$ – mantysa (*significand*)
- p – dokładność

Przez znormalizowaną liczbę zmiennopozycyjną rozumie się liczbę, którą można przedstawić w powyższej postaci.[4]

[4] Posługiwanie się znormalizowną reprezentacją powoduje, że nie można w niej przedstawić liczby 0! Zwykle przedstawia się ją jako $1.0 \times \beta^{e_{\min}-1}$.

Liczby zmiennopozycyjne

Nie każdą liczbę rzeczywistą można przedstawić dokładnie w postaci zmiennopozycyjnej. Np. 0.1 ma w dwójkowej reprezentacji nieskończone rozwinięcie i leży między dwiema liczbami zmiennopozycyjnymi.

Swoich reprezentacji zmiennopozycyjnych nie mają liczby większe niż $\beta \times \beta^{e_{\max}}$ i mniejsze niż $1.0 \times \beta^{e_{\min}}$.

Rozkład liczb zmiennopozycyjnych na osi liczbowej nie jest jednorodny![5]

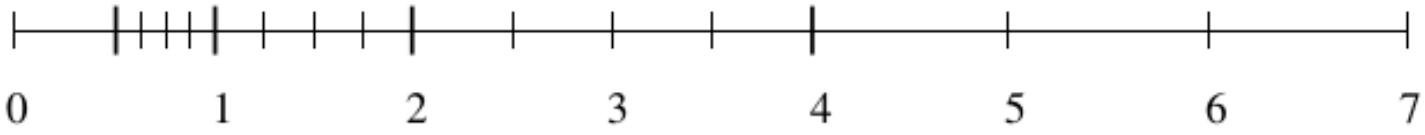


Figure D-1 Normalized numbers when β = 2, p = 3, e_{min} = -1, e_{max} = 2

[5] D. Goldberg, *What every computer scientist should know about floating-point arithmetic?*, Computing Surveys, March 1991

Liczby zmiennopozycyjne

Liczba bitów kodująca liczby zmiennopozycyjne

$$\log_2(e_{\max} - e_{\min} + 1) + \log_2(\beta^p) + 1$$

Operacje wykonywane na liczbach zmiennopozycyjnych są obarczone błędami zaokrągleń.

Przykład: Niech $\beta = 10$ i $p = 3$.
Jeśli w wyniku obliczeń otrzymujemy wynik 3.12×10^{-2}, a wynik dokładny wynosi 0.0314, to błąd wynosi 2 jednostki na ostatnim miejscu (jom).

Jeśli liczba rzeczywista 0.0314159 jest reprezentowana jako 3.14×10^{-2}, to błąd wynosi 0.159 jom.

Liczby zmiennopozycyjne

Jeśli liczba zmiennopozycyjna $d.ddd\ldots d\beta^e$ reprezentuje z, to błąd wynosi

$$|d.ddd\ldots d - z/\beta^e|\beta^{p-1}\mathrm{jom}$$

Jeśli mamy lzp najbliższą prawdziwego wyniku, to może być ona obarczona błędem 0.5 jom.

Innym sposobem oceny poprawności przybliżenia liczby rzeczywistej przez lzp jest wyznaczanie błędu względnego.

Liczby zmiennopozycyjne

Jaki błąd względny odpowiada błędowi 0.5 jom?

Jeśli przybliżamy liczbę rzeczywistą przez jej zmiennopozycyjną reprezentację $\tilde{x} = d.ddd\ldots d\beta^e$, to możemy popełnić maksymalnie błąd $0.000\ldots0\beta' \times \beta^e$, gdzie $\beta' = \beta/2$ i mamy p cyfr mantysy liczby oraz $p-1$ zer w mantysie błędu. Błąd wynosi $(\beta/2)\beta^{-p}\beta^e$.

Ponieważ $\beta^e \leqslant \tilde{x} \leqslant \beta\beta^e$, więc

$$\frac{1}{2}\beta^{-p} \leqslant \frac{1}{2}\text{jom} \leqslant \frac{1}{2}\beta\beta^{-p}$$

Błąd względny odpowiadający 0.5 jom może zmieniać się o czynnik β.

Ten czynnik określa się mianem kołysania (*wobble*).

Liczby zmiennopozycyjne

Liczba rzeczywista zaokrąglona do najbliższej lzp jest obarczona błędem względnym ograniczonym przez wielkość $\varepsilon = (\beta/2)\beta^{-p}$.

Określenia ε:

- epsilon maszynowy
- (względna) dokładność maszynowa
- jednostka zaokrąglenia

Jeśli lzp ma błąd n jom, to oznacza, że $\log_\beta(n)$ jej cyfr jest zanieczyszczonych. Podobnie jest, jeśli błąd względny wynosi $n\varepsilon$.

Przy szacowaniu błędów wprowadzanych przez rozmaite wzory wygodniej jest posługiwać się błędami względnymi.[6]

[6] Jeśli wystarczy ocena rzędu wielkości błędu, to można do tego celu używać zarówno jom-u jaki i błędu względnego, gdyż różnią się one co najwyżej o czynnik β.

Liczby zmiennopozycyjne: odejmowanie

Jednym ze sposobów wyznaczania różnicy dwóch lzp jest wykonanie działania dokładnie i zaokrągleniu wyniku do najbliższej lzp. Jest to kosztowne, jeśli liczby różnią się znacznie wielkością.

Przykład:

$$\begin{aligned} x &= 2.150000000000000000 \times 10^{12} \\ y &= 0.000000000000000125 \times 10^{12} \\ x - y &= 2.149999999999999875 \times 10^{12} \end{aligned}$$

Po zaokrągleniu: 2.15×10^{12}. Jeśli operacje wykonać w arytmetyce zmiennopozycyjnej z $p = 3$, to

$$\begin{aligned} x &= 2.15 \times 10^{12} \\ y &= 0.00 \times 10^{12} \\ x - y &= 2.15 \times 10^{12} \end{aligned}$$

Liczby zmiennopozycyjne: odejmowanie

Przykład: Niech $x = 10.1$, $y = 9.93$.

$$\begin{aligned} x &= 1.01 \times 10^1 \\ y &= 0.99 \times 10^1 \\ x - y &= 0.02 \times 10^1 \end{aligned}$$

Poprawny wynik wynosi 0.17, więc błąd różnicy wynosi 0.03=3 jom. Błąd względny (0.03/0.17=0.18) jest równy 35ε ($\varepsilon = 0.005$) i tylko jedna cyfra wyniku jest poprawna!

Tw. W arytmetyce zmiennopozycyjnej z parametrami β i p wyznaczanie różnicy może być obarczone błędem względnym $\leqslant \beta - 1$.

Jeśli $\beta = 2$, to błąd może być tak duży jak wynik. Dla $\beta = 10$ – błąd względny może być 9 razy większy od wyniku!

Ponieważ $(\beta - 1)/\varepsilon = \beta^p$, więc przy błędzie względnym rzędu β wszystkie cyfry wyniku są zanieczyszczone.

Liczby zmiennopozycyjne: cyfry chroniące (*guard digits*)

Co się dzieje, jeśli operacja odejmowania zostanie przeprowadzona z użyciem dodatkowej cyfry (cyfry chroniącej)?

$$\begin{aligned} x &= 1.010 \times 10^1 \\ y &= 0.993 \times 10^1 \\ x - y &= 0.017 \times 10^1 \end{aligned}$$

Wynik jest dokładny!

Przy użyciu jednej cyfry chroniącej otrzymuje się wyniki, które są obarczone błędem względnym, który może być nieco większy niż ε.

Wniosek: Bez zastosowania cyfry chroniącej odejmowanie od siebie bliskich sobie liczb (lub dodawanie wielkości o przeciwnych znakach) jest obarczone dużymi błędami względnymi, gdyż cyfry znaczące obu składników są identyczne i się znoszą.

Znoszenie się cyfr znaczących może mieć charakter katastrofalny lub łagodny.

Liczby zmiennopozycyjne: znoszenie się cyfr znaczących

Katastrofalne znoszenie się cyfr znaczących – składniki wyrażenia są podatne na błędy zaokrąglenia.

Przykład: Wyznaczyć $\sqrt{b^2 - 4ac}$ dla $b = 3.34$, $a = 1.22$ i $c = 2.28$.

Łagodne znoszenie się cyfr znaczących – odejmowanie dotyczy dokładnie znanych wielkości.

Jeśli to odejmowanie wykonywane jest przy użyciu cyfry chroniącej, to błąd nie powinien być większy niż 2ε.

Wzory, które wykazują katastrofalne znoszenie się cyfr mogą często być zapisane w innej postaci, która będzie wolna od tego problemu.

Przykład: Wyznaczanie pierwiastków trójmianu kwadratowego.
Przykład: Wyznaczanie $x^2 - y^2$ dla $x \approx y$.

Liczby zmiennopozycyjne: działania arytmetyczne

Wyniki działań arytmetycznych nie muszą być lzp (liczbami maszynowymi), więc są obarczone błędami zaokragleń. Jeśli

$$\begin{aligned} x +^* y &\stackrel{\text{df}}{=} \mathrm{fl}(x + y) \\ x -^* y &\stackrel{\text{df}}{=} \mathrm{fl}(x - y) \\ x \times^* y &\stackrel{\text{df}}{=} \mathrm{fl}(x \times y) \\ x \ /^* y &\stackrel{\text{df}}{=} \mathrm{fl}(x \ / \ y) \end{aligned}$$

to

$$\begin{aligned} x +^* y &= (x + y)(1 + \varepsilon_1) \\ x -^* y &= (x - y)(1 + \varepsilon_2) \\ x \times^* y &= (x \times y)(1 + \varepsilon_3) \\ x \ /^* y &= (x \ / \ y)(1 + \varepsilon_4) \end{aligned}$$

gdzie $\varepsilon_i \leqslant \varepsilon$.

Wynik działań zmiennopozycyjnych można traktować jako wynik dokładnych działań wykonanych na zaburzonych danych wejściowych.

Liczby zmiennopozycyjne: działania arytmetyczne

Jeśli $\beta = 10$, $p = 7$ i

$$\begin{aligned} x &= 0.12345670 \times 10^0 \\ y &= 0.4 \times 10^{-7} \\ z &= 0.4 \times 10^{-7} \\ x + y + z &= 0.12345678 \times 10^0 \end{aligned}$$

to

$$\begin{aligned} \mathrm{fl}(x + y) &= 0.1234567 \times 10^0 \\ \mathrm{fl}((x + y) + z) &= 0.1234567 \times 10^0 \\ \mathrm{fl}(y + z) &= 0.8 \times 10^{-7} \\ \mathrm{fl}(y + z) &= 0.0000001 \times 10^0 \\ \mathrm{fl}(x + (y + z)) &= 0.1234568 \times 10^0 \end{aligned}$$

Dlatego np. $\mathrm{fl}\left(\Sigma_{n=1}^{N} n^{-2}\right) \neq \mathrm{fl}\left(\Sigma_{n=N}^{1} n^{-2}\right)$.

Liczby zmiennopozycyjne: działania arytmetyczne

Dodawanie lzp nie jest działaniem łącznym!

Tw. (wzór summacyjny Kahana)

Jeśli $\Sigma_{i=1}^{N} x_i$ jest obliczona wg następującego algorytmu

```
s=x(1)
c=0
do i=2,N
  y=x(i)-c
  t=s+y
  c=(t-s)-y
  s=t
enddo
```

to wyznaczona suma równa się

$$\sum_{i=1}^{N} x_i(1+\delta_i) + O(N\varepsilon^2)\sum |x_i|, \quad |\delta_i| \leqslant 2\varepsilon$$

Standard IEEE 754

- IEEE 754 – $\beta = 2$, $p = 24$ (pojedyncza precyzja), $p = 53$ (podwójna precyzja), dokładnie określone ułożenie bitów w obu formatach liczb

  zalety: małe błędy względne i małe kołysanie, ukryta jedynka mantysy daje dodatkowy bit precyzji (zero musi być kodowane jako $1.0 \times 2^{e_{\text{min}}-1}$)

- IEEE 854 – $\beta = 2$ lub 10, wymagania, co do dozwolonych wartości p

format	p	e_{min}	e_{max}	e width	format width
single	24	-126	127	8	32
single-extended	32	$\leqslant$-1022	1023	$\leqslant$11	43
double	53	-1022	1023	11	64
double-extended	64	$\leqslant$-16382	$>$16383	15	79

Standard IEEE 754

- Wykładnik liczby (*unbiased exponent*) jest zapisywany jako liczba całkowita bez znaku, tzn. wynosi $e - 127$ (SP) lub $e - 1023$ (DP). e to wykładnik z przesunięciem (*biased exponent*).
- Operacje dodawania, odejmowania, mnożenia, dzielenia, pierwiastkowania, dzielenia z resztą, zamiany liczb całkowitych na z.p. (i odwrotnie) są dokładnie zaokrąglane, tzn. wyniki tych działań są liczone dokładnie, a potem zaokrąglane do parzystej
- Dokładne zaokrąglanie nie dotyczy operacji zamiany liczb dwójkowych na dziesiętne i odwrotnie (brak efektywnych algorytmów), wyznaczania wartości funkcji przestępnych.
- Definiuje liczby zdenormalizowane, zero ze znakiem, NaN, Inf.

Specyfikacja wyników operacji arytmetycznych zapewnia przenaszalność programów oraz umożliwia dowodzenie poprawności algorytmów.

Standard IEEE 754: specjalne wartości

exponent	fraction	value	name
$e = e_{\text{min}} - 1$	$f = 0$	± 0	signed zero
$e = e_{\text{min}} - 1$	$f \neq 0$	$0.f \times 2^{e_{\text{min}}}$	denormalized
$e_{\text{min}} \leqslant e \leqslant e_{\text{max}}$	any	$1.f \times 2^{e}$	normalized
$e = e_{\text{max}} + 1$	$f = 0$	∞	infinity
$e = e_{\text{max}} + 1$	$f \neq 0$	NaN	Not A Number

Standard IEEE 754

- NaN – tradycyjnie wyrażenia $0/0$ lub $\sqrt{-1}$ powodują przerwanie wykonania programu; często warto przetwarzanie kontynuować.
  - Standard nie precyzuje dokładnej postaci liczby NaN, tzn. w zależności od implementacji tych liczb (niezerowa) mantysa może zawierać informacje zależne od systemu.
  - Jeśli w jakimś wyrażeniu wystąpi liczba NaN wraz z lzp, to wynik będzie NaN.
  - NaN – działania dające w wyniku NaN

działanie	wyrażenie
$+$	$\infty + (-\infty)$
$\times$	$0 \times \infty$
$/$	$0/0 \quad \infty/\infty$
REM	x REM 0, ∞ REM y
$\sqrt{}$	$\sqrt{x}, \quad x < 0$

Standard IEEE 754

- Inf (∞) – obliczenia są kontynuowane, kiedy pojawia się nadmiar
  - jest to bezpieczniejsze, niż zastępowanie wyniku największą możliwą liczbą
  - $0/0$ zwraca NaN, ale $\pm 1/0 = \pm\infty$
  - żeby określić wynik wyrażenia, w którym występuje ∞ wystarczy zastąpić ją wielkością skończoną i przejść do granicy:

    $$3/0 = +\infty, \quad 4 - \infty = -\infty, \quad \sqrt{\infty} = \infty$$

    Jeśli granica nie istnieje, to dostajemy NaN.
  - jeśli w wyrażeniu występuje ∞, to wynikiem może być zwykła liczba, gdyż $x/\infty = 0$

Standard IEEE 754

- ± 0

  Standard IEEE definiuje, że $+0 = -0$, a nie $-0 < +0$.

  Zalety

  – operacje typu $3 \cdot (+0) = +0$, $+0/-3 = -0$ dają poprawne wyniki
  – wyrażenia $1/(1/x) = x$ są spełnione dla $x = \pm\infty$
  – możliwość obsługi funkcji posiadających nieciągłość w zerze, np. $\log x$
  – ułatwienia w realizacji arytmetyki liczb zespolonych, np. trzeba zapewnić, że $\sqrt{1/z} = 1/\sqrt{z}$

Standard IEEE 754

- Liczby zdenormalizowane – jeśli $e = e_{\min}$, to nie wymaga się, aby liczba była znormalizowana

  Problem: dla $\beta = 2$ mamy ukryty bit, więc mantysa jest zawsze większa od 1.0.

Bity mantysy	$b_1, b_2, \ldots, b_{p-1}$	
Wykładnik	$e > e_{\min} - 1$	$e = e_{\min} - 1$
Liczba	$1.b_1 b_2 \ldots b_{p-1} \times 2^e$	$0.b_1 b_2 \ldots b_{p-1} \times 2^{e_{\min}}$

Niech $\beta = 10$, $p = 3$ i $e_{\min} = -98$, $x = 6.78 \times 10^{-97}$, $y = 6.81 \times 10^{-97}$.

$x, y < 1.0 \times 10^{-98}$, $(x - y) \neq 0$, ale fl$(x - y) = 0$

$x - y = 0.06 \times 10^{-97} = 6.00 \times 10^{-99}$ nie jest lzp

Musi być zastąpiona przez 0!

Standard IEEE 754

- Liczby zdenormalizowane

  Jak ważne jest zachowanie własności $x = y \iff x - y = 0$?

  Jak powinien zachowywać się fragment kodu

  ```
  if (x <> y) then z=1/(x-y)?
  ```

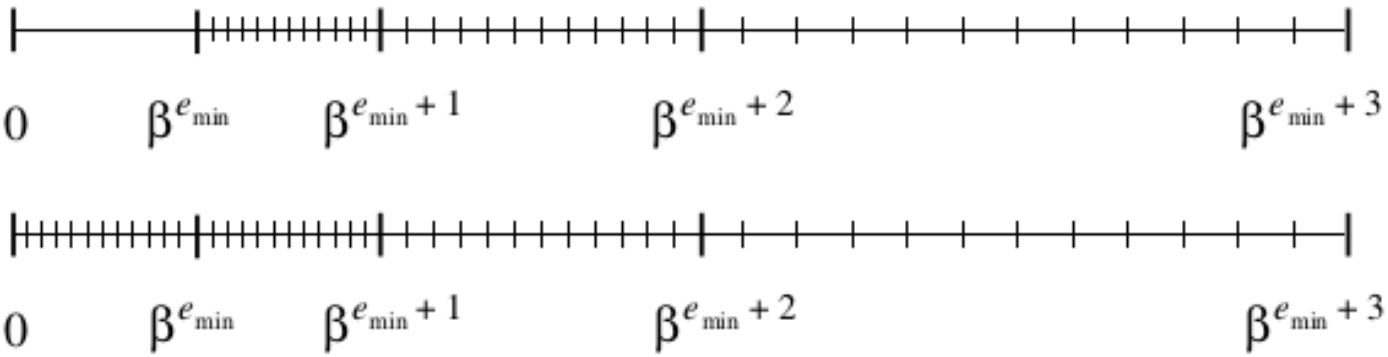
  Utrzymanie własności $x = y \iff x - y = 0$ dla wszystkich lzp ułatwia pisanie i sprawdzanie poprawności programów.

  Standard IEEE zapewnia, że tego typu własności są spełnione przez stosowanie zdenormalizowanych lzp.

  Uwaga! Wystąpienie niedomiaru powoduje 16-krotne spowolnienie wykonywania operacji.

Standard IEEE 754: łagodny niedomiar

Użycie liczb zdenormalizowanych pozwala na realizację tzw. stopniowego niedomiaru (*gradual underflow*).



Łagodna obsługa niedomiaru powoduje, że gęstośc małych lzp nie zmienia się gwałtownie, jeśli wyniki obliczeń stają się mniejsze niż $\beta^{e_{\min}}$.

Algorytmy, które wykazują stosunkowo duże błędy względne przy wykonywaniu obliczeń na liczbach bliskich niedomiarowi zachowują się poprawnie w tym zakresie.

Standard IEEE 754: Wyjątki, flagi, procedury obsługi wyjątków

- Klasy wyjątków: nadmiar (*overflow*), niedomiar (*underflow*), dzielenie przez zero (*division by zero*), niepoprawna operacja (*invalid operation*) i niedokładna operacja (*inexact operation*).

  Dla każdej klasy jest przydzielona oddzielna flaga statusu.

- gfortran określa pułapki:
  `overflow, underflow, zero, invalid, inexact/precision.`[7]

- Jeśli pojawia się wyjątek w postaci dzielenia przez zero, nadmiaru, niedomiaru, itp. to ustala się wynik operacji i kontynuuje obliczenia.

- Jeśli pojawia się wyjątek to podnoszona jest flaga statusu; użytkownicy powinni mieć możliwość czytania i ustawiania tej flagi.[8]

- Standard zaleca, aby były dostępne procedury obsługi wyjątków.

[7] Niektóre architektury procesora oferują także pułapkę `denormal`.

[8] Flagi są typu *sticky*, tzn. jeśli zostały ustawione to muszą być *explicite* opuszczone. Tylko badając flagę można odróżnić $1/0$, który daje ∞ od nadmiaru.

Układy równań liniowych

Równania

$$\sum_{j=1}^{n} a_{ij}x_i = b_i, \quad a_{ij}, b_i \in \mathcal{R}, \quad i = 1, \ldots, n$$

definiują układ n równań liniowych o n niewiadomych.

W postaci macierzowej

$$\boldsymbol{Ax} = \boldsymbol{b}$$

$$\begin{bmatrix} a_{11} \ldots a_{1n} \\ a_{21} \ldots a_{2n} \\ \vdots \ldots \vdots \\ a_{n1} \ldots a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

$\boldsymbol{A} = [a_{ij}]$ (macierz główna układu), $\boldsymbol{A_b} = [a_{ij}|b_i]$ (macierz uzupełniona)
$\boldsymbol{x} = [x_1 \ldots x_n]^T$, $\boldsymbol{b} = [b_1 \ldots b_n]^T$

Układy równań liniowych

Wyróżnia się układy o macierzy $\boldsymbol{A}$

- pełnej (mało elementów równych zeru) i niedużej ($n < 10^3 - 10^4$)

  metody rozwiązywania: dokładne/bezpośrednie; przy braku błędów zaokrągleń rozwiązanie otrzymuje się po skończonej liczbie kroków

- rzadkiej (mało elementów różnych od zera) i dużej

  metody rozwiązywania: iteracyjne, specjalizowane metody bezpośrednie

Macierze osobliwe i nieosobliwe

Macierz $\boldsymbol{A}$ jest osobliwa, jeśli

- nie posiada macierzy odwrotnej, tj. nie istnieje $\boldsymbol{M}$ taka, że $\boldsymbol{AM} = \boldsymbol{MA} = \boldsymbol{I}$
- $\det(\boldsymbol{A}) = 0$[9]
- $\mathrm{r}(\boldsymbol{A}) < n$, tj. maksymalna liczba liczba liniowo niezależnych wierszy lub kolumn jest mniejsza niż n

  $\mathrm{r}(\boldsymbol{A}) = n \iff \det(\boldsymbol{A}) \neq 0$
- $\boldsymbol{Az} = \boldsymbol{o}$ dla jakiegś $\boldsymbol{z} \neq \boldsymbol{o}$, $\boldsymbol{o} = [0 \ldots 0]^T$

Jeśli macierz $\boldsymbol{A}$ jest nieosobliwa, to istnieje $\boldsymbol{A^{-1}}$ i $\boldsymbol{x} = \boldsymbol{A^{-1}b}$ dla dowolnego wektora $\boldsymbol{b}$; $\mathrm{r}(\boldsymbol{A}) = r(\boldsymbol{A}|\boldsymbol{b}) = n$.

Jeśli $\det(\boldsymbol{A}) = 0$, to układ równań może nie mieć rozwiązań (układ sprzeczny) lub mieć wiele rozwiązań (jeśli $\boldsymbol{x}$ jest rozwiązaniem, to $\boldsymbol{x} + \gamma\boldsymbol{z}$ też jest rozwiązaniem dla dowolnego skalara γ).

[9] Dla $n = 1$, $\det(\boldsymbol{A}) = a_{11}$. Dla $n > 1$, $\det(\boldsymbol{A}) = \sum_{i=1}^{n}(-1)^{i+n}a_{in}\det(\boldsymbol{A}_{in})$, gdzie $\boldsymbol{A}_{in}$ powstaje z $\boldsymbol{A}$ po wykreśleniu i-tego wiersza i n-tej kolumny.

Przykład 1: $\det(\boldsymbol{A}) \neq 0$

$$\begin{cases} 2x_1 + 3x_2 = b_1 \\ 5x_1 + 4x_2 = b_2 \end{cases}$$

$$x_1 = -\frac{1}{7}\begin{vmatrix} b_1 & 3 \\ b_2 & 4 \end{vmatrix} = -\frac{1}{7}(4b_1 - 3b_2)$$

$$x_2 = -\frac{1}{7}\begin{vmatrix} 2 & b_1 \\ 5 & b_2 \end{vmatrix} = -\frac{1}{7}(-5b_1 + 2b_2)$$

$\det(\boldsymbol{A}) = -7$: rozwiązanie istnieje dla każdego wektora $\boldsymbol{b}$.

Proste

$$x_2 = -\frac{2}{3}x_1 + \frac{b_1}{3} \qquad x_2 = -\frac{5}{4}x_1 + \frac{b_2}{4}$$

przecinają się w jednym punkcie.

Przykład 2: $\det(\boldsymbol{A}) = 0$

$$\begin{cases} 2x_1 + 3x_2 = b_1 \\ 4x_1 + 6x_2 = b_2 \end{cases} \iff \begin{cases} 2x_1 + 3x_2 = b_1 \\ 0x_1 + 0x_2 = b_2 - 2b_1 \end{cases}$$

Jeśli $\boldsymbol{b} = [4\ 7]^T$, to sprzeczność, bo $0 \neq -1$. Brak rozwiązań!

Jeśli $\boldsymbol{b} = [4\ 8]^T$, to $\boldsymbol{x} = [\gamma \quad (4-2\gamma)/3]^T$. Nieskończenie wiele rozwiązań.

Układ równań

$$\begin{cases} 2x_1 + 3x_2 = b_1 \\ 5x_1 + 4x_2 = b_2 \end{cases}$$

można traktować jako dwa równania prostych w układzie x_1, x_2.

Proste mogą się przecinać dokładnie w jednym punkcie ($\det(\boldsymbol{A}) \neq 0$), mogą się pokrywać lub mogą być równoległe ($\det(\boldsymbol{A}) = 0$).

Jeśli $\boldsymbol{a}_1 = [2\ 5]^T$, $\boldsymbol{a}_2 = [3\ 4]^T$, to

$$x_1\boldsymbol{a}_1 + x_2\boldsymbol{a}_2 = \boldsymbol{b}$$

Jeśli istnieje jednoznaczne rozwiązanie tego układu, to znaczy, że wektor $\boldsymbol{b}$ można rozłożyć na sumę dwóch innych wektorów $x_1\boldsymbol{a_1}$ i $x_2\boldsymbol{a_2}$. Jest to możliwe, jeśli wektory $\boldsymbol{a_1}$ i $\boldsymbol{a_2}$ są liniowo niezależne, czyli wtedy i tylko wtedy, gdy $\det(\boldsymbol{A}) \neq 0$.

Formalnie układ równań

$$\sum_{j=1}^{n} a_{ij}x_i = b_i, \quad i = 1, \ldots, n$$

ma rozwiązania dane przez wzory Cramera

$$x_k = \frac{1}{\det(\boldsymbol{A})} \begin{vmatrix} a_{11} & a_{12} & \ldots & a_{1k-1} & b_1 & a_{1k+1} & \ldots & a_{1n} \\ a_{21} & a_{22} & \ldots & a_{2k-1} & b_2 & a_{2k+1} & \ldots & a_{2n} \\ \vdots & \vdots & \ldots & \vdots & \vdots & \vdots & \ldots & \vdots \\ a_{n1} & a_{n2} & \ldots & a_{nk-1} & b_n & a_{nk+1} & \ldots & a_{nn} \end{vmatrix}$$

Metoda wymaga wyznaczenia $n + 1$ wyznaczników, czyli wykonania $2p_n(n + 1)!$ mnożeń i dodawań, gdzie

$$\lim_{n\to\infty} p_n = \lim_{n\to\infty} \sum_{i=2}^{n} \frac{1}{(i-1)!} = e - 1$$

Dla $n = 15$ oznacza to wykonanie około 7.24×10^{13} operacji, czyli około 10 godz. CPU o wydajności 2GFLOPS!

Układy równań liniowych o macierzach trójkątnych

Układy postaci

$$\boldsymbol{U}\boldsymbol{x} = \boldsymbol{b}$$

$$\begin{bmatrix} u_{11} & u_{12} & \dots & u_{1n-1} & u_{1n} \\ 0 & u_{22} & \dots & u_{2n-1} & u_{2n} \\ \vdots & \vdots & \dots & \vdots & \vdots \\ 0 & 0 & \dots & u_{nn-1} & u_{nn} \\ 0 & 0 & \dots & 0 & u_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_{n-1} \\ b_n \end{bmatrix}$$

rozwiązuje się metodą podstawiania wstecz.

Jeśli $u_{ii} \neq 0,\ i = 1, 2, \dots, n$, to

$$x_{nn} = b_n/u_{nn}, \quad x_i = \frac{1}{u_{ii}} \left(b_i - \sum_{k=i+1}^{n} u_{ik} x_k \right), \quad i = n-1, \dots, 1$$

Układy równań liniowych o macierzach trójkątnych

Układ z macierzą trójkątną dolną rozwiązuje się analogicznie

$$\boldsymbol{L}\boldsymbol{x} = \boldsymbol{b}$$

$$x_{11} = b_1/l_{11}, \quad x_i = \frac{1}{l_{ii}}\left(b_i - \sum_{k=1}^{i-1} l_{ik}x_k\right), \quad i = 2, \ldots, n$$

Rozwiązanie układu trójkątnego wymaga

$$n + 2\sum_{i=2}^{n}(i-1) = n + 2\sum_{i=1}^{n-1} i = n + (n-1)n = n^2$$

operacji zmiennopozycyjnych: n dzieleń oraz $n(n-1)/2$ mnożeń i odejmowań.

Eliminacja Gaussa

- Układ równań liniowych można przekształcić w układ równoważny dodając do dowolnego równania kombinację liniową innych równań.
- Jeśli $\det(\boldsymbol{A}) \neq 0$, to można przekształcić macierz układu w macierz trójkątną górną.
- Rozwiązanie uzyskuje się przez podstawianie wstecz.

Eliminacja Gaussa

Jeśli macierz $\boldsymbol{M_k} = [m_{ij}]$ ma elementy

$$m_{ij} = \begin{cases} 1 & i = j \\ m_{ik} = a_{ik}/a_{kk}, & i = k+1, \ldots, n \\ 0 & \text{pozostałe przypadki} \end{cases}$$

i $\boldsymbol{a_k}$ jest k-tą kolumną macierzy $\boldsymbol{A}$, to $\boldsymbol{M_k a_k}$ daje wektor z wyzerowanymi elementami a_{ik}, $i = k+1, \ldots, n$.

$$\boldsymbol{M_k a_k} = \begin{bmatrix} 1 & \ldots & 0 & 0 & \ldots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \ldots & 1 & 0 & \ldots & 0 \\ 0 & \ldots & -m_{k+1k} & 1 & \ldots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \ldots & -m_{nk} & 0 & \ldots & 1 \end{bmatrix} \begin{bmatrix} a_{1k} \\ \vdots \\ a_{kk} \\ a_{k+1k} \\ \vdots \\ a_{nk} \end{bmatrix} = \begin{bmatrix} a_{1k} \\ \vdots \\ a_{kk} \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

Eliminacja Gaussa

Własności $\boldsymbol{M}$

- $\det(\boldsymbol{M}) \neq 0$
- $\boldsymbol{M_k} = \boldsymbol{I} - \boldsymbol{m_k}\boldsymbol{e_k}^T$
  $\boldsymbol{m_k} = [0, \ldots, m_{k+1k}, \ldots, m_{nk}]^T$
  $\boldsymbol{e_k} = [0, \ldots, 1, \ldots, 0]^T$ – k-ta kolumna macierzy jednostkowej
- $\boldsymbol{M_k^{-1}} = \boldsymbol{I} + \boldsymbol{m_k}\boldsymbol{e_k}^T = \boldsymbol{L_k}$
- $k > j$

$$\begin{aligned}
\boldsymbol{M_k M_j} &= (\boldsymbol{I} - \boldsymbol{m_k}\boldsymbol{e_k}^T)(\boldsymbol{I} - \boldsymbol{m_j}\boldsymbol{e_j}^T) \\
&= \boldsymbol{I} - \boldsymbol{m_k}\boldsymbol{e_k}^T - \boldsymbol{m_j}\boldsymbol{e_j}^T + \boldsymbol{m_k}\boldsymbol{e_k}^T\boldsymbol{m_j}\boldsymbol{e_j}^T \\
&= \boldsymbol{I} - \boldsymbol{m_k}\boldsymbol{e_k}^T - \boldsymbol{m_j}\boldsymbol{e_j}^T + \boldsymbol{m_k}\boldsymbol{o}\boldsymbol{e_j}^T \\
&= \boldsymbol{I} - \boldsymbol{m_k}\boldsymbol{e_k}^T - \boldsymbol{m_j}\boldsymbol{e_j}^T
\end{aligned}$$

Eliminacja Gaussa $O(n^4)$

Niech $\boldsymbol{A}_1 = \boldsymbol{A}$, $\boldsymbol{b}_1 = \boldsymbol{b}$.

Eliminację Gaussa wykonuje się w $n-1$ krokach:

1: $\boldsymbol{M}_1\boldsymbol{A}_1\boldsymbol{x} = \boldsymbol{M}_1\boldsymbol{b}_1,\ \boldsymbol{A}_2\boldsymbol{x} = \boldsymbol{b}_2$

⋮

k: $\boldsymbol{M}_k\boldsymbol{A}_k\boldsymbol{x} = \boldsymbol{M}_k\boldsymbol{b}_k,\ \boldsymbol{A}_{k+1}\boldsymbol{x} = \boldsymbol{b}_{k+1}$

⋮

n-1: $\boldsymbol{M}_{n-1}\boldsymbol{A}_{n-1}\boldsymbol{x} = \boldsymbol{M}_{n-1}\boldsymbol{b}_{n-1},\ \boldsymbol{A}_n\boldsymbol{x} = \boldsymbol{b}_n$

Wynikowy układ równań o macierzy trójkątnej górnej rozwiązuje się poprzez podstawianie wstecz

$$\boldsymbol{A}_n\boldsymbol{x} = \boldsymbol{U}\boldsymbol{x} = \boldsymbol{b}_n$$

Eliminacja Gaussa

Układy $\boldsymbol{A}_k\boldsymbol{x} = \boldsymbol{b}_k$ i $\boldsymbol{A}_{k+1}\boldsymbol{x} = \boldsymbol{b}_{k+1}$ są formalnie równoważne.

Ponieważ $\det(\boldsymbol{M}_k) \neq 0$, więc jeśli $\det(\boldsymbol{A}_k) \neq 0$, to

$$\det(\boldsymbol{A}_{k+1}) = \det(\boldsymbol{M}_k\boldsymbol{A}_k) = \det(\boldsymbol{M}_k)\det(\boldsymbol{A}_k) \neq 0$$

Zatem, jeśli układ $\boldsymbol{A}_k\boldsymbol{x} = \boldsymbol{b}_k$ ma jednoznaczne rozwiązanie, to układ $\boldsymbol{A}_{k+1}\boldsymbol{x} = \boldsymbol{b}_{k+1}$ także je posiada.

Ponieważ

$$\boldsymbol{M}_{n-1}\ldots\boldsymbol{M}_1\boldsymbol{A} = \boldsymbol{U}$$

więc jeśli w trakcie eliminacji wyznaczać także $\boldsymbol{L}_k$, to otrzymujemy rozkład macierzy $\boldsymbol{A}$ na czynniki trójkątne

$$\boldsymbol{A} = \boldsymbol{M}_1^{-1}\ldots\boldsymbol{M}_{n-1}^{-1}\boldsymbol{U} = \boldsymbol{L}_1\ldots\boldsymbol{L}_{n-1}\boldsymbol{U} = \boldsymbol{L}\boldsymbol{U}$$

Eliminacja Gaussa $O(2n^3/3)$

Wektor $\boldsymbol{b}$ podlega przekształcaniu tak samo jak kolumny macierzy $\boldsymbol{A}$.

Niech $\boldsymbol{A}_k = [a_{ij}^{(k)}]$ i $a_{in+1}^{(k)} = b_i^{(k)}, \quad (1 \leqslant k \leqslant i \leqslant n)$.

Eliminację wykonuje się w $n-1$ krokach $k = 1, 2, \ldots, n-1$.

W k-tym kroku element $a_{ij}^{(k)}$, dla $j > k$ przekształca się wg wzorów

$$a_{ij}^{(k+1)} = a_{ij}^{(k)} - m_{ik} a_{kj}^{(k)}, \quad m_{ik} = a_{ik}^{(k)} / a_{kk}^{(k)}$$

dla $i = k+1, \ldots, n, j = k+1, \ldots, n+1$.

Eliminacja Gaussa wymaga wykonania

$$\sum_{k=1}^{n-1} [2(n-k)(n-k+p) + (n-k)] + pn^2 \approx \frac{2}{3} n^3$$

operacji arytmetycznych, gdzie p jest liczbą wektorów $\boldsymbol{b}$.

Jak duże może być n, żeby metoda eliminacji Gaussa mogła być zastosowana?

Eliminacja Gaussa: wybór elementów głównych

W każdym kroku element główny $a_{kk}^{(k)}$ musi być różny od zera.

Przykład: Poniższy układ równań jest nieosobliwy:

$$\begin{cases} x_1 + x_2 + x_3 = 1 \\ x_1 + x_2 + 2x_3 = 2 \\ x_1 + 2x_2 + 2x_3 = 1 \end{cases}$$

Pierwszy krok eliminacji Gaussa daje:

$$\begin{cases} x_1 + x_2 + x_3 = 1 \\ x_3 = 1 \\ x_2 + x_3 = 0 \end{cases}$$

Kolejny krok można wykonać pod warunkiem zamiany wierszy 2 i 3.

Wniosek: Dowolny układ nieosobliwy można zredukować do postaci trójkątnej stosując eliminację Gaussa wraz z przestawianiem wierszy.

Eliminacja Gaussa: wybór elementów głównych

Przykład: $a_{kk}^{(k)} \approx 0$

$$\begin{cases} x_1 + & x_2 + & x_3 = 1 \\ x_1 + 1.0001x_2 + & 2x_3 = 2 \\ x_1 + & 2x_2 + 2x_3 = 1 \end{cases}$$

Pierwszy krok eliminacji Gaussa powoduje, że $a_{22}^{(1)} = 0.0001$.

Podstawianie wstecz:

p	x_1	x_2	x_3
4	0	0.0	1.000
5	0.9999	-1.0000	1.0001
16	1.0	-1.00010001	1.00010001

Eliminacja Gaussa: wybór elementów głównych

Eliminacja Gaussa powinna być łączona z częściowym lub pełnym wyborem elementów głównych.

Wybór elementów głównych nie jest potrzebny, jeśli

- macierz $\boldsymbol{A}$ ma dominującą główną przekątną, tj.

$$|a_{ii}| > \sum_{j=1, i\neq j}^{n} |a_{ij}|, \quad i = 1, 2, \ldots$$

- $\boldsymbol{A}^T = \boldsymbol{A}$ i $\boldsymbol{x}^T \boldsymbol{A} \boldsymbol{x} > 0$

Jeśli współczynniki układu równań różnią się od siebie o wiele rzędów wielkości, to układ taki trzeba przed eliminacją poddać normalizacji, tj. każde równanie musi zostać przeskalowane czynnikiem $1/\max_j |a_{ij}|$, $i = 1, \ldots, n$.

Eliminacja Gaussa: rozkład $\boldsymbol{LU}$

Tw. Niech $\boldsymbol{A}$ będzie macierzą $n \times n$. Niech $\boldsymbol{A}_k$ oznacza macierz utworzoną z elementów początkowych pierwszy k wierszy i kolumn macierzy $\boldsymbol{A}$. Jeśli $\det(\boldsymbol{A}_k) \neq 0$, dla $k = 1, \ldots n$, to istnieje jedyny rozkład $\boldsymbol{A} = \boldsymbol{LU}$ na czynniki trójkątne takie, że macierz $\boldsymbol{L} = [m_{ij}]$ jest macierzą trójkątną dolną z $m_{ii} = 1$, $i = 1, \ldots n$, a macierz $\boldsymbol{U}$ jest macierzą trójkątną górną.

Równość $\boldsymbol{A} = \boldsymbol{LU}$ oznacza

$$a_{ij} = \sum_{p=1}^{r} m_{ip} u_{pj}, \quad r = \min(i, j), \quad i, j = 1, \ldots, n$$

Jeśli $m_{ii} = 1$, to jest to układ n^2 równań na n^2 niewiadomych będących elementami macierzy $\boldsymbol{L}$ i $\boldsymbol{U}$.

Schematy zwarte: metoda Doolittle'a i Crouta

W n krokach wyznacza się elementy kolejnych wierszy $\boldsymbol{U}$ i kolum $\boldsymbol{L}$ wg równań

$$a_{kj} = \sum_{p=1}^{k} m_{kp} u_{pj}, \quad j \geqslant k$$

$$a_{ik} = \sum_{p=1}^{k} m_{ip} u_{pk}, \quad i > k$$

czyli

$$u_{kj} = a_{kj} - \sum_{p=1}^{k-1} m_{kp} u_{pj}, \quad j = k, k+1, \ldots, n$$

$$m_{ik} = \frac{1}{u_{kk}} \left(a_{ik} - \sum_{p=1}^{k-1} m_{ip} u_{pk} \right), \quad i = k+1, \ldots, n$$

Powyższe wzory określają zwarty wariant metody Gaussa, tzw. metodę Doolittle'a. Przy założeniu $u_{ii} = 1$ otrzymuje się metodę Crouta. Można je łatwo łączyć z częściowym wyborem elementów głównych.

Metoda Choleskiego

Jeśli macierz układu jest symetryczna i dodatnio określona, to $\boldsymbol{U} = \boldsymbol{L}^T$, więc $u_{kk} = m_{kk}$ i $u_{pk} = m_{kp}$ i wzory metody Doolittle'a przyjmują postać

$$
\begin{aligned}
m_{kk} &= \left(a_{kk} - \sum_{p=1}^{k-1} m_{kp}^2 \right)^{1/2} \\
m_{ik} &= \frac{1}{m_{kk}} \left(a_{ik} - \sum_{p=1}^{k-1} m_{ip} m_{kp} \right), \quad i = k+1, \ldots, n
\end{aligned}
$$

Jest to tzw. metodę Choleskiego (pierwiastków kwadratowych), która wiąże się ściśle z symetrycznym wariantem eliminacji Gaussa.

Macierz odwrotna

Ponieważ $\boldsymbol{A}\boldsymbol{A}^{-1} = \boldsymbol{I}$, więc poszczególne kolumny macierzy $\boldsymbol{A}^{-1}$ można wyznaczyć rozwiązując n układów równań liniowych

$$\boldsymbol{A}\boldsymbol{x}_k = \boldsymbol{e}_k, \quad k = 1, \ldots, n$$

gdzie $\boldsymbol{e}_k$ jest k-tą kolumną macierzy jednostkowej.

Koszt: $8n^3/3$.[10]

Łatwiej to zrobić obliczając $\boldsymbol{A}^{-1}$ jako $\boldsymbol{U}^{-1}\boldsymbol{L}^{-1}$, gdyż macierze odwrotne wyznacza się łatwo przez podstawianie wstecz

$$\boldsymbol{L}\boldsymbol{y}_k = \boldsymbol{e}_k, \qquad \boldsymbol{U}\boldsymbol{z}_k = \boldsymbol{e}_k, \quad k = 1, \ldots, n$$

Koszt: $2n^3/3 + 2n^3/6 + n^3/3 = 4n^3/3$.

Przykład: Jak wyznaczyć $\boldsymbol{A}^{-1}\boldsymbol{B}$?

[10] Dla p wektorów $\boldsymbol{b}$ koszt eliminacji Gaussa wynosi $2n^3/3 + 2pn^2$.

Wyznacznik macierzy

Z równoważności eliminacji Gaussa i rozkładu na czynniki trójkątne oraz definicji wyznacznika wynika, że

$$\det(\boldsymbol{A}) = \det(\boldsymbol{LU}) = \det(\boldsymbol{L}) \det(\boldsymbol{U}) = (-1)^q a_{11}^{(1)} a_{22}^{(2)} \cdots a_{nn}^{(n)}$$

gdzie, q określa liczbę przestawień wierszy lub kolumn macierzy w trakcie wykonywania eliminacji.

Specjalne układy równań liniowych

Jeśli macierz układu równań liniowych ma specyficzne własności, to można je zwykle wykorzystać przy opracowaniu specjalnych wersji algorytmów rozwiązywania takich układów.

Wyróżnia się macierze

- symetryczne, tj. $\boldsymbol{A} = \boldsymbol{A}^T$
- dodatnio określone, tj. $\boldsymbol{x}^T \boldsymbol{A}\boldsymbol{x} > \boldsymbol{o}$ dla $\boldsymbol{x} \neq \boldsymbol{0}$
- wstęgowe
- rzadkie

Analiza błędów

Dokładność rozwiązania układu równań liniowych jest zależna od dokładności elementów macierzy $\boldsymbol{A}$ oraz wektora $\boldsymbol{b}$, a także od wielkości błędów zaokrągleń.

Jeśli $\tilde{\boldsymbol{x}}$ jest obliczonym rozwiązaniem układu $\boldsymbol{Ax} = \boldsymbol{b}$, to residuum definiujemy jako

$$\boldsymbol{r} = \boldsymbol{b} - \boldsymbol{A}\tilde{\boldsymbol{x}}$$

Jeśli $\boldsymbol{r} = \boldsymbol{0}$, to $\tilde{\boldsymbol{x}} = \boldsymbol{A}^{-1}\boldsymbol{b}$. Jeśli $\boldsymbol{r}$ małe, to wektor $\tilde{\boldsymbol{x}}$ powinien być dobrym rozwiązaniem.

Przykład (wg Kahana): Jeśli

$$\begin{cases} 1.2969x_1 + 0.8648x_2 = 0.8642 \\ 0.2161x_1 + 0.1441x_2 = 0.1440 \end{cases}$$

to wektor $\tilde{\boldsymbol{x}} = [0.9911, -04870]^T$ daje residuum $\boldsymbol{r} = [-10^{-8}, 10^{-8}]^T$, ale prawdziwe rozwiązanie wynosi $\boldsymbol{x} = [2, -2]^T$!

Analiza błędów: normy wektorów i macierzy

Dla wektorów definiuje się normy

- $\|\boldsymbol{x}\|_1 = \Sigma_{i=1}^{n} |x_i|$
- $\|\boldsymbol{x}\|_2 = \left(\Sigma_{i=1}^{n} |x_i|^2\right)^{1/2}$ (euklidesowa)
- $\|\boldsymbol{x}\|_\infty = \max_{1 \leqslant i \leqslant n} |x_i|$ (maksymalna)

Własności normy:

- $\|\boldsymbol{x}\| > 0$ dla $\boldsymbol{x} \neq 0$; $\|\boldsymbol{x}\| = 0 \iff \boldsymbol{x} = \mathbf{0}$
- $\|\gamma \boldsymbol{x}\| = |\gamma| \|\boldsymbol{x}\|$, γ dowolne
- $\|\boldsymbol{x} + \boldsymbol{y}\| \leqslant \|\boldsymbol{x}\| + \|\boldsymbol{y}\|$ (nierówność trójkąta)

Normy macierzowe definiuje się w oparciu o normy wektorowe. Jeśli

$$\|\boldsymbol{A}\boldsymbol{x}\| \leqslant \|\boldsymbol{A}\| \, \|\boldsymbol{x}\|$$

to normy wektora i macierzowa są zgodne.

Analiza błędów: normy wektorów i macierzy

Normy wektorowe indukują zgodne z nimi normy macierzowe:

$$\|\boldsymbol{A}\| = \max_{\boldsymbol{x}\neq\boldsymbol{0}} \frac{\|\boldsymbol{A}\boldsymbol{x}\|}{\|\boldsymbol{x}\|}$$

$$\|\boldsymbol{A}\|_1 = \max_{1\leqslant j\leqslant n} \sum_{i=1}^{n} |a_{ij}| \qquad \|\boldsymbol{A}\|_\infty = \max_{1\leqslant i\leqslant n} \sum_{j=1}^{n} |a_{ij}|$$

Własności

- $\|\boldsymbol{A}\| > 0$ dla $\boldsymbol{A} \neq \boldsymbol{O}$
- $\|\gamma\boldsymbol{A}\| = |\gamma|\|\boldsymbol{A}\|$, γ dowolne
- $\|\boldsymbol{A}+\boldsymbol{B}\| \leqslant \|\boldsymbol{A}\| + \|\boldsymbol{B}\|$
- $\|\boldsymbol{A}\boldsymbol{B}\| \leqslant \|\boldsymbol{A}\|\,\|\boldsymbol{B}\|$ (nierówność Schwarza)
- $\|\boldsymbol{A}\boldsymbol{x}\| \leqslant \|\boldsymbol{A}\|\,\|\boldsymbol{x}\|$, $\boldsymbol{x}$ dowolny

Analiza błędów: wskaźnik uwarunkowania dla macierzy

Jeśli $\boldsymbol{A}(\boldsymbol{x}+\delta\boldsymbol{x}) = \boldsymbol{b}+\delta\boldsymbol{b}$,to $\|\boldsymbol{b}\| \leqslant \|\boldsymbol{A}\|\,\|\boldsymbol{x}\|$ i $\|\delta\boldsymbol{x}\| \leqslant \|\boldsymbol{A}^{-1}\|\,\|\delta\boldsymbol{b}\|$, więc

$$\frac{\|\delta\boldsymbol{x}\|}{\|\boldsymbol{x}\|} \leqslant \|\boldsymbol{A}\|\,\|\boldsymbol{A}^{-1}\|\frac{\|\delta\boldsymbol{b}\|}{\|\boldsymbol{b}\|} = \mathrm{cond}(\boldsymbol{A})\|\frac{\|\delta\boldsymbol{b}\|}{\|\boldsymbol{b}\|}$$

Jeśli $(\boldsymbol{A}+\delta\boldsymbol{A})(\boldsymbol{x}+\delta\boldsymbol{x}) = \boldsymbol{b}$, to $\|\delta\boldsymbol{x}\| \leqslant \|\boldsymbol{A}^{-1}\|\,\|\delta\boldsymbol{A}\|\,\|\boldsymbol{x}+\delta\boldsymbol{x}\|$, więc

$$\frac{\|\delta\boldsymbol{x}\|}{\|\boldsymbol{x}+\delta\boldsymbol{x}\|} \leqslant \|\boldsymbol{A}\|\,\|\boldsymbol{A}^{-1}\|\frac{\|\delta\boldsymbol{A}\|}{\|\boldsymbol{A}\|} = \mathrm{cond}(\boldsymbol{A})\frac{\|\delta\boldsymbol{A}\|}{\|\boldsymbol{A}\|}$$

$\mathrm{cond}(\boldsymbol{A})$ to wskaźnik uwarunkowania macierzy $\boldsymbol{A}$, który określa jak blisko macierzy do macierzy osobliwej.[11]

[11] Dla macierzy osobliwej $\mathrm{cond}(\boldsymbol{A}) = \infty$.

Analiza błędów: wskaźnik uwarunkowania dla macierzy

Jeśli wskaźnik uwarunkowania macierzy jest duży, to małe względne zaburzenia macierzy $\boldsymbol{A}$ i wektora $\boldsymbol{b}$ powodują duże względne zaburzenia wektora $\boldsymbol{x}$, więc zadanie rozwiązania układu $\boldsymbol{Ax} = \boldsymbol{b}$ jest źle uwarunkowane.

Przykład: Dla macierzy

$$\boldsymbol{A} = \begin{bmatrix} 1.2969 & 0.8648 \\ 0.2161 & 0.1441 \end{bmatrix}$$

$$\boldsymbol{A}^{-1} = 10^8 \begin{bmatrix} 0.1441 & -0.8648 \\ -0.2161 & 1.2969 \end{bmatrix}$$

wskaźnik uwarunkowania $\|\boldsymbol{A}\| \, \|\boldsymbol{A}^{-1}\|$ wynosi około $2.1617 \times 1.5130 \times 10^8 \approx 3.3 \times 10^8$.

Można pokazać, że błędy zaokrągleń prowadzą do takiego zaburzenia rozwiązania, że

$$\|\delta \boldsymbol{x}\| \leqslant 2\varepsilon \operatorname{cond}(\boldsymbol{A}) \|\boldsymbol{x}\|_\infty$$

Aproksymacja – sformułowanie zagadnienia

Problem: W jaki sposób najlepiej przybliżyć (aproksymować) funkcję $f(x)$ za pomocą elementu $f^*(x)$ należącego do pewnej klasy funkcji?

O funkcjach tej klasy zakłada się, że

- można na nich łatwo wykonywać operacje matematyczne (wielomiany, funkcje wymierne lub trygonometryczne)
- każda z nich zależy od wartości liczbowych pewnych parametrów

Aproksymacja jest specjalnym przypadkiem ogólniejszego problemu dostosowania modelu matematycznego do określonych danych i innych znanych faktów.

Problemy aproksymacji wiążą się także ze statystyką matematyczną.

Aproksymacja – sformułowanie zagadnienia

Niedokładności aproksymacji:

- dane wejściowe są (najczęściej) obarczone błędami (błędy pomiarowe)
- wybór klasy funkcji uwarunkowany zastosowanym modelem (błędy obcięcia)

Aproksymacja liniowa – sformułowanie zagadnienia

Klasa funkcji aproksymujących jest postaci

$$f^*(x) = \sum_{k=1}^{m} c_k \phi_k(x),$$

$\phi_k(x)$ są ustalonymi funkcjami, a c_k, $k = 1, \ldots, m$ są współczynnikami, które trzeba wyznaczyć.

Przykład: Jeśli $\phi_k(x) = x^k$, to klasą dopuszczalnych funkcji są wielomiany stopnia co najwyżej m. Układ $\{1, x, \ldots, x^m\}$ nazywa się bazą zbioru wszystkich wielomianów stopnia m.

Zakładamy, że funkcja aproksymowana jest dana w postaci tablicy swoich wartości $f(x_i)$, $i = 1, \ldots, n$ na siatce $G = \{x_i\}_{i=1}^n$.

Aproksymacja liniowa – sformułowanie zagadnienia

Chcemy dobrać wartości m parametrów c_i, $i = 1, \ldots, m$ tak, aby spełnione były równości

$$f^*(x_i) = f(x_i), \quad i = 1, \ldots, n$$

czyli musi być spełniony następujący układ n równań z m niewiadomymi

$$\sum_{k=1}^{m} c_k \phi_k(x_i) = f(x_i), \quad i = 1, \ldots, n$$

Jeśli $m = n$, to (zwykle) taki układ ma dokładnie jedno rozwiązanie i f^* jest określone przez interpolację (kolokację).

Przykład: Interpolacja liniowa: $m = 2$, $\phi_1(x) = 1$, $\phi_2(x) = x$.

Warunkiem jednoznaczności rozwiązania jest niezależność liniowa wektorów

$$\boldsymbol{\phi}_k = [\phi_k(x_1)\phi_k(x_2) \ldots \phi_k(x_n)]^T$$

tzn. funkcje $\phi_k(x)$ muszą być liniowo niezależne na siatce, czyli macierz zbudowana z wektorów $\boldsymbol{\phi}_k$ musi być nieosobliwa.

Aproksymacja liniowa – sformułowanie zagadnienia

Jeśli $n > m$ to zwykle

$$\sum_{k=1}^{m} c_k \phi_k(x_i) \approx f(x_i), \quad i = 1, \ldots, n$$

W przypadku układu nadokreślonego musimy się (zwykle) zadowolić przybliżonym spełnieniem równań.

Aproksymacja polega na takim wyborze parametrów c_k, żeby funkcja błędu $y = f^* - f$ była jak najmniejsza (w sensie jakiejś normy).

Normy/seminormy funkcji

- norma maksymalna:

$$\|f\|_\infty = \max_{x\in[a,b]} |f(x)|$$

- norma L_2 (Euklidesowa):

$$\|f\|_2 = \left(\int_a^b |f(x)|^2 dx\right)^{1/2}$$

- norma ważona L_2 (Euklidesowa ważona):

$$\|f\|_{2,w} = \left(\int_a^b w(x)|f(x)|^2 dx\right)^{1/2}$$

  Funkcja wagowa $w(x)$ musi być ciągła i dodatnia w przedziale (a, b) (na końcach przedziału dopuszcza się osobliwości).

- seminorma ważona (dla danej siatki G)[12]

$$\|\mathrm{tab}(f)\|_2 = \left(\sum_{i=1}^{m} w(x_i)|f(x_i)|^2\right)^{1/2}$$

[12] Z tego, że $\|tab(f)\|_2 = 0$ nie wynika, że $f(x) \equiv 0$ tożsamościowo (funkcja znika zazwyczaj tylko w węzłach).

Aproksymacja – sformułowanie zagadnienia

Metody aproksymacji opierają się na zasadzie minimalizacji pewnej (semi)normy funkcji błędu $y = f^* - f$.

Wartości funkcji na pewnej siatce punktów można traktować jako wektor w pewnej przestrzeni liniowej. Im ten wektor ma więcej składowych, tym lepiej przybliża funkcję f.

Problem aproksymacji sprowadza się do znalezienia funkcji (wektora) f^* leżącego najbliżej funkcji (wektora) f.

Aproksymacja liniowa: m liniowo niezależnychi funkcji, które tworzą bazę m-wymiarowej przestrzeni liniowej. Poszukjemy takiej kombinacji liniowej tych funkcji, aby odległość pomiędzy funkcjami f i f^* była jak najmniejsza.

Ta odległość jest równa długości wektora $f^* - f$, który jest prostopadły do podprzestrzeni liniowej napiętej przez funkcje bazowe.

Prostopadłość wektorów – iloczyn skalarny

Określenie prostopadłości wektorów w przestrzeni liniowej wymaga wprowadzenia pojęcia iloczynu skalarnego

$$(f,g) = \begin{cases} \int_a^b w(x)f(x)g(x)dx & \text{przypadek ciągły} \\ \sum_{i=1}^{n} w(x_i)f(x_i)g(x_i) & \text{przypadek dyskretny} \end{cases}$$

Własności:

- $(f,g) = (g,f)$
- $(c_1 f_1 + c_2 f_2, g) = c_1(f_1,g) + c_2(f_2,g)$
- $(f,f) = \|f\|^2 \geqslant 0$

gdzie $\|\cdot\|$ oznacza normę euklidesową w przypadku ciągłym i seminormę ważoną w przypadku dyskretnym.

Funkcje f i g są ortogonalne, jeśli $(f,g) = 0$.

Aproksymacja średniokwadratowa – sformułowanie zadania

Niech f będzie funkcją ciągłą, którą należy przybliżyć w przedziale (a, b) za pomocą kombinacji liniowej

$$f^*(x) = \sum_{k=1}^{m} c_i \phi_k(x)$$

Zadanie aproksymacji średniokwadratowej polega na wyznaczeniu współczynników c_i w taki sposób, aby Euklidesowa (semi)norma ważona była jak najmniejsza. c_k minimalizują

$$\|f^* - f\|^2 = \int_a^b w(x)|f^*(x) - f(x)|^2 dx$$

lub

$$\|f^* - f\|^2 = \sum_{i=0}^{n} w(x_i)|f^*(x_i) - f(x_i)|^2$$

Rozwiązanie tego zagadnienia daje metoda najmniejszych kwadratów.

Aproksymacja średniokwadratowa – rozwiązanie

Tw. Jeśli funkcje ϕ_k, $k = 1, \ldots, m$ są liniowo niezależne, to zagadnienie aproksymacji średniokwadratowej ma jedyne rozwiązanie

$$f^* = \sum_{k=1}^{m} c_k^* \phi_k,$$

gdzie współczynniki c_k^* spełniają tzw. równanie normalne

$$\sum_{k=1}^{m} c_i^* (\phi_i, \phi_k) = (f, \phi_k), \qquad k = 1, \ldots, m.$$

Dla wszystkich funkcji bazowych $(f^* - f, \phi_k) = 0$, $k = 1, \ldots, m$

Jeśli $(\phi_i, \phi_k) = \delta_{ik}$, to $c_k^* = (f, \phi_k)/(\phi_k, \phi_k)$.

c_k^* – współczynniki ortogonalne (Fouriera)

Równania normalne

Równania normalne wynikają z minimalizowania wyrażenia

$$\begin{aligned}\|f^* - f\|^2 &= (\sum_i c_i\phi_i - f, \sum_j c_j\phi_j - f) \\ &= \sum_{ij} c_i c_j(\phi_i, \phi_j) + (f, f) - 2\sum_i c_i(\phi_i, f)\end{aligned}$$

$\|f^* - f\|^2$ osiąga minimum, gdy $\delta\|f^* - f\|^2 = 0$, tzn.

$$\frac{\partial}{\partial c_k}\|f^* - f\|^2 = 0, \qquad k = 1, \ldots, m$$

$$\sum_i c_i(\phi_i, \phi_k) = (f, \phi_k), \qquad k = 1, \ldots, m$$

Równania normalne

Przykład: Należy użyć metody najmniejszych kwadratów do aproksymacji funkcją $f^* = c_1 + c_2 x$ następujących danych pomiarowych

x	1	3	4	6	7
y	-2.1	-0.9	-0.6	0.6	0.9

Przy założeniu, że $w(x_i) = 1$, $i = 1, \ldots, 5$, otrzymujemy układ równań normalnych

$$\begin{cases} 5c_1 + \ 21c_2 = -2.1 \\ 21c_1 + 111c_2 = \ \ 2.7 \end{cases}$$

którego rozwiązaniem jest $c_1 = -2.542$ i $c_2 = 0.5053$.

$$\text{tab}(f^* - f) = [0.063, -0.126, 0.079, -0.110, 0.095]^T$$

$(\phi_1, f^* - f) = 0.001$, $(\phi_2, f^* - f) = 0.006$.

Gdyby nie błędy zaokrągleń, to te iloczyny skalarne powinny być równe zeru.

Równania normalne – wartość średnia

Przykład: Niech $m = 1$ i $\phi_1(x) = 1$ i niech

$$(f, g) = \sum_{i=1}^{n} w_i f(x_i) g(x_i)$$

W takim przypadku układ równań redukuje się do jednego równania

$$c_1(\phi_1, \phi_1) = (\phi_1, f)$$

$$c_1 = \frac{(\phi_1, f)}{(\phi_1, \phi_1)} = \frac{\Sigma_{i=1}^{n} w_i \phi_1(x_i) f(x_i)}{\Sigma_{i=1}^{n} w_i \phi_1(x_i) \phi_1(x_i)} = \frac{\Sigma_{i=1}^{n} w_i f(x_i)}{\Sigma_{i=1}^{n} w_i}$$

Współczynnik c_1 nazywa się średnią ważoną wartości f. Gdy $w_i = w$, $i = 1, \ldots, n$ to

$$c_1 = \frac{1}{n} \Sigma_{i=1}^{n} f(x_i)$$

Wniosek: Wyznaczanie średniej jest szczególnym przypadkiem aproksymacji średniokwadratowej.

Regresja liniowa

Problem: Należy określić parametry a i b modelu

$$y^* = f^*(x; a, b) = a + bx$$

które będzie opisywał N punktów (x_i, f_i), każdy obarczony błędem pomiarowym σ_i (wartości x_i znamy dokładnie).

Parametry muszą być tak dobrane, żeby zminimalizować

$$\|\mathrm{tab}(y) - \mathrm{tab}(y^*(x; a, b))\|^2$$

$$\chi^2 = \sum_{i=1}^{n} \left(\frac{y_i - y^*(x_i; a, b)}{\sigma_i} \right)^2 = \sum_{i=1}^{n} \left(\frac{y_i - a - bx_i}{\sigma_i} \right)^2$$

Regresja liniowa

Warunki minimum

$$0 = \frac{\partial \chi^2}{\partial a} = -2 \sum_{i=1}^{n} \frac{(y_i - a - bx_i)}{\sigma_i^2}$$

$$0 = \frac{\partial \chi^2}{\partial b} = -2 \sum_{i=1}^{n} \frac{x_i(y_i - a - bx_i)}{\sigma_i^2}$$

Jeśli

$$S \equiv \sum_{i=1}^{n} 1/\sigma_i^2, \qquad S_x \equiv \sum_{i=1}^{n} x_i/\sigma_i^2 \qquad S_y \equiv \sum_{i=1}^{n} y_i/\sigma_i^2$$

$$S_{xx} \equiv \sum_{i=1}^{n} x_i^2/\sigma_i^2 \qquad S_{xy} \equiv \sum_{i=1}^{n} x_i y_i/\sigma_i^2$$

to

$$\begin{cases} aS + bS_x = S_y \\ aS_x + bS_{xx} = S_{xy} \end{cases}$$

Regresja liniowa

Jeśli $\Delta \equiv SS_{xx} - (S_x)^2$, to

$$a = \frac{1}{\Delta}\left(S_{xx}S_y - S_x S_{xy}\right) \qquad b = \frac{1}{\Delta}\left(SS_{xy} - S_x S_y\right)$$

Jeśli błędy poszczególnych pomiarów są niezależne od siebie, to[13]

$$\sigma_a^2 = \sum_{i=1}^{n} \sigma_i^2 \left(\frac{\partial a}{\partial y_i}\right)^2 = S_{xx}/\Delta \qquad \sigma_b^2 = \sum_{i=1}^{n} \sigma_i^2 \left(\frac{\partial b}{\partial y_i}\right)^2 = S/\Delta$$

Kowariancja i korelacja parametrów:

$$\mathrm{Cov}(a,b) = -S_x/\Delta, \qquad r_{ab} = -S_{xx}/\sqrt{SS_{xx}}$$

Dobroć dopasowania modelu dana jest przez niezupełną funkcję γ:

$$Q\left(\frac{n-2}{2}, \frac{\chi^2}{2}\right)$$

[13] Jeśli nie znamy błędów σ_i indywidualnych pomiarów, to $\sigma_i \equiv 1$. Wzory na σ_a^2 i σ_b^2 trzeba przemnożyć przez $\sqrt{\chi^2/(n-2)}$.

Modelowanie danych

Problem: Jak zwięźle opisać dane doświdczalne/obserwacyjne?

Jeśli rozumiemy przebieg zjawiska, to możemy stworzyć model, który je opisuje i który powinien odtwarzać dane doświadczalne.

Aproksymacja średniokwadratowa: modelem jest zbiór funkcji bazowych.

Dopasowanie dostarcza zbioru parametrów $a_1, \ldots, a_M$ modelu, które dobiera się tak, aby minimalizować funkcję błędu

$$\|f^*(x; a_1, \ldots, a_m) - f(x)\|^2 = \begin{cases} \int_a^b w(x)(f^*(x; a_1, \ldots, a_m) - f(x))^2 dx \\ \sum_{i=1}^n w(x_i)(f^*(x_i; a_1, \ldots, a_m) - f(x_i))^2 dx \end{cases}$$

Zagadnienie dopasowywania parametrów jest równoważne minimalizacji w wielu wymiarach.

Wyodrębnia sie przypadek modelowania, gdyż pozwala on zastosować lepsze, efektywniejsze metody dopasowywania (optymalizacji) parametrów.

Modelowanie danych

Trudność modelowania: dane doświadczalne są obarczone błędami, są zaszumione, więc nigdy nie pasują do modelu, nawet dokładnego.

Zadania modelowania:

- określenie dokładności wyznaczonych parametrów
- ocena, czy model jest odpowiedni do posiadanych danych; wymaga to porównania *dobroci dopasowania* względem pewnego użytecznego standardu statystycznego
- ocena jakości nie szczególnego dopasowanie, lecz uzyskanie pewności, że w innej cześci przestrzeni parametrów nie istnieje lepsze dopasowanie.

Modelowanie danych

Proces optymalizacji jest użyteczny jeśli dostarcza

- optymalne parametry
- błędy tych parametrów
- statystyczną ocenę (miarę) jakości (dobroci) dopasowania

Jeśli statystyczna miara dopasowania jest niezadawalająca, to dane odnośnie parametrów i ich błędów są bezwartościowe.

Modelowanie danych

- prawdziwe parametry zna Natura
- prawdziwe parametry prowadzą do wyników, które są realizowane z przypadkowymi błędami; są to zmierzone dane $D_{(0)}$
- dane $D_{(0)}$ prowadzą do zestawu parametrów $a_{(0)}$
- $D_{(0)}$ nie jest jedyną realizacją prawdziwych parametrów $a_{(p)}$ (błędy przypadkowe)
- istnieje nieskończenie wiele innych realizacji tych parametrów w postaci tzw. zbiorów danych hipotetycznych $D_{(1)}$, $D_{(2)}$, ...

Modelowanie danych

- każdy zbiór danych hipotetycznych prowadzi do innego zbioru parametrów $a_{(1)}$,$a_{(2)}$, ...; pojawiają się one zgodnie z pewnym rozkładem prawdopodobieństwa w m-wymiarowej przestrzeni wszystkich możliwych zbiorów tych parametrów
- zmierzony zbiór $a_{(0)}$ jest jednym z członków rodziny zbiorów parametrów.
- rozkład $a_{(i)} - a_{(p)}$ zawiera (ilościową) informację o niepewnościach tkwiących w doświadczeniu, ale jest nieznany
- zakładamy, że rozkłady $a_{(i)} - a_{(p)}$ oraz $a_{(i)} - a_{(0)}$ niewiele się od siebie różnią, co pozwala wyznaczać przedziały ufności odnośnie otrzymywanych wartości

Metoda najmniejszych kwadratów – uzasadnienie

W metodzie najmniejszych kwadratów dopasowujemy do n znanych punktów (x_i, f_i) model, który posiada m parametrów

$$f^*(x) = f^*(x; a_1, \ldots, a_m)$$

Parametry dobieramy tak, aby wyrażenie

$$\sum_{i=1}^{m} \left(f_i - f^*(x_i; a_1, \ldots, a_m)\right)^2 w_i$$

osiągało minimum względem a_j, $j = 1, \ldots m$.

Dlaczego tak można robić? Z jakiej zasady wynika to wyrażenie?

Metoda najmniejszych kwadratów – estymator największej wiarygodności/szansy (*maximum likelihood estimator*).

Metoda najmniejszych kwadratów – uzasadnienie

Dla danego zbioru punktów (x_i, f_i) możemy określić wiele różnych zbiorów parametrów $a_{(i)}$?

Intuicja podpowiada, że niektóre są lepsze, bardziej prawdopodobne niż inne.

Lepsze parametry to te parametry, które dobrze odtwarzają (przybliżają) dane.

W jaki sposób opisać ilościowo to przeświadczenie?

Czy istnieje odpowiedź na pytanie:

> Jakie jest prawdopodobieństwo, że dany zestaw parametrów $\{a_j\}$, $j = 1, \ldots, m$, jest poprawny?

Metoda najmniejszych kwadratów – uzasadnienie

Pytanie poprawne:

> Jakie jest prawdopodobieństwo, że przy danym zbiorze parametrów pojawi się pewien zbiór danych, czyli wartości $f_i^* \pm \Delta f_i^*$?

Małe prawdopodobieństwo – parametry dalekie od poprawnych

Duże prawdopodobieństwo – parametry poprawne

Wniosek: Utożsamiamy prawdopodobieństwo wystąpienia pewnych danych przy założonych parametrach z szansą pojawienia się dobrych parametrów dla danych punktów.

Metoda najmniejszych kwadratów – uzasadnienie[14]

Prawdopodobieństwo zaobserwowania x-sukcesów w ciągu N doświadczeń Bernoulliego z prawdopodobieństwem sukcesu θ każdego z nich, wynosi

$$p_\theta(x) = \binom{N}{x}\theta^x(1-\theta)^{N-x}, \qquad x \in X = \{1, 2, \ldots, N\}$$

Przy ustalonym x

$$L(\theta; x) = \binom{N}{x}\theta^x(1-\theta)^{N-x}, \qquad \theta \in [0, 1]$$

pozwala określić wiarygodność postulowanego parametru θ.

Jeśli

$$L(\theta_2; x) > L(\theta_1; x)$$

to dla danej wartości x zaobserwowanie wyniku jest bardziej prawdopodobne przy $\theta = \theta_2$ niż $\theta = \theta_1$.

[14] R. Zielinski, *Siedem wykładów wprowadzających do statystyki matematycznej*, Biblioteka Matematyczna, tom 72, PWN

Metoda najmniejszych kwadratów – uzasadnienie

Zasada największej wiarygodności

Jeśli wynikiem obserwacji jest x, to za estymator nieznanej wartości parametru θ należy przyjąć tę wartość θ, która maksymalizuje wiarygodność $L(\theta; x)$.

Zasada największej wiarygodności wynika z intuicji.

Optymalne parametry pozyskujemy maksymalizując estymator wiarygodności.

Metoda najmniejszych kwadratów – uzasadnienie

Jeśli wyniki pomiarów są od siebie niezależne, a każdy pomiar jest obarczony błędem o rozkładzie normalnym ze średnią $f^*(x_i)$ i wariancją σ_i^2, to prawdopodobieństwo wystąpienia wartości f_i wynosi

$$\frac{1}{\sqrt{2\pi}\sigma_i} \exp\left(-\frac{(f_i - f^*(x_i))^2}{2\sigma_i^2}\right) \Delta f$$

gdzie $|f_i - f^*(x_i)| < \Delta f$.

Prawdopodobieństwo otrzymania n punktów f_i wynosi

$$P = \prod_i^n \left\{\exp\left(-\frac{(f_i - f^*(x_i))^2}{2\sigma_i^2}\right) \Delta f\right\}$$

Maksymalizowanie tego wyrażenia jest równoważne maksymalizowaniu jego logarytmu lub minimalizowaniu ujemnej wartości tego logarytmu

$$\sum_{i=1}^n \left(\frac{f_i - f^*(x_i)}{\sqrt{2}\sigma_i}\right)^2 - n \log \Delta f \Longrightarrow \sum_{i=1}^n \left(\frac{f_i - f^*(x_i)}{\sigma_i}\right)^2$$

Metoda najmniejszych kwadratów – aproksymacja χ^2

Niech zmienne losowe χ_i mają rozkład normalny ze średnią $\bar{\chi}_i$ i wariancją σ_i^2. Zmienne losowe

$$Z_i = \frac{\chi_i - \bar{\chi}i}{\sigma_i}$$

są zmiennymi losowymi o rozkładzie normalnym o średniej równej zeru i jednostkowej wariancji.

Jeśli $Z_1, Z_2, \ldots, Z_k$ są niezależnymi zmiennymi losowymi o standardowym rozkładzie normalnym, to

$$\chi^2 = \sum_{i=1}^{k} Z_i^2$$

jest zmienną losową χ_k^2 o rozkładzie zwanym rozkładem chi-kwadrat o k-stopniach swobody.

Metoda najmniejszych kwadratów – aproksymacja χ^2

W przypadku aproksymacji średniokwadratowej minimalizujemy wyrażenia

$$\chi^2 = \sum_{i=1}^{n} \left(\frac{f_i - f^*(x_i; a_1, \ldots, a_m)}{\sigma_i} \right)^2$$

Rozkład różnych wartości χ^2 w okolicach minimum może być przybliżony analitycznym rozkładem chi-kwadrat dla $n - m$ stopni swobody.

Wielkość

$$Q\left(\frac{n-m}{2}, \frac{\chi^2}{2}\right)$$

określa prawdopodobieństwo, że rozkład chi-kwadrat przyjmie przez przypadek wartość większą niż χ^2.

Metoda najmniejszych kwadratów – aproksymacja χ^2

Jeśli dla jakiegoś zestawu danych wartość Q jest bardzo małym prawdopodobieństwem, to

- model jest zły i może być statystycznie odrzucony
- faktyczne błędy pomiarowe są większe niż założone
- błędy pomiarowe nie mają rozkładu normalnego

Q stanowi pewną miarę dobroci dopasowania (*goodness-of-fit*).

Reguła wynikająca z praktyki powiada, że dla względnie dobrego dopasowania $\chi^2 \approx n - m$.

Wyznaczając pochodną funkcji χ^2 względem kolejnych parametrów i przyrównując je do zera otrzymujemy układ m równań (na m niewiadomych a_i), które muszą być spełnione w minimum funkcji χ^2.

Ogólna liniowa metoda najmniejszych kwadratów

Do punktów (x_i, y_i) należy dopasować model

$$y(x) = \sum_{k=1}^{m} a_k X_k(x)$$

gdzie $X_1(x), \ldots, X_m(x)$ są dowolnymi ustalonymi funkcjami bazowymi (te funkcje nie muszą być liniowe!). Liniowa jest zależność modelu od jego parametrów.

Parametry a_k dobieramy tak, aby zminimalizować funkcję

$$\chi^2 = \sum_{i=1}^{n} \left(\frac{y_i - \Sigma_{k=1}^{m} a_k X_k(x_i)}{\sigma_i} \right)^2$$

σ_i są błędami standardowymi poszczególnych pomiarów (jeśli nie są znane to kładziemy $\sigma_i = 1$).

Niech $A_{ij} = \frac{X_j(x_i)}{\sigma_i}$, $b_i = \frac{y_i}{\sigma_i}$

$$\boldsymbol{A} = \begin{bmatrix} \frac{X_1(x_1)}{\sigma_1} & \frac{X_2(x_1)}{\sigma_1} & \cdots & \frac{X_m(x_1)}{\sigma_1} \\ \frac{X_1(x_2)}{\sigma_2} & \frac{X_2(x_2)}{\sigma_2} & \cdots & \frac{X_m(x_2)}{\sigma_2} \\ \vdots & \vdots & \cdots & \vdots \\ \frac{X_1(x_n)}{\sigma_n} & \frac{X_2(x_n)}{\sigma_n} & \cdots & \frac{X_m(x_n)}{\sigma_n} \end{bmatrix}$$

Macierz wzorcowa ma (zwykle) więcej wierszy niż kolumn, gdyż musimy dysponować większą liczbą danych niż parametrów, które próbujemy wyznaczyć. Minimalizacja funkcji χ^2 prowadzi do równań normalnych zagadnienia najmniejszych kwadratów

$$(\boldsymbol{A^T A})\boldsymbol{a} = \boldsymbol{A^T b}$$

$$\boldsymbol{\alpha a} = \boldsymbol{\beta}$$

Taki układ można rozwiązać np. przez rozkład na czynniki trójkątne plus podstawianie wstecz lub eliminację Gaussa.

Można pokazać, że

$$\sigma^2(a_j) = C_{jj} = (\alpha^{-1})_{jj}, \qquad \mathrm{Cov}(a_i, a_j) = C_{ij} = (\alpha^{-1})_{ij}$$

Diagonalne elementy macierzy $\boldsymbol{C}$ są wariancjami dopasowywanych parametrów, a pozadiagonalne – kowariancjami między poszczególnymi parami różnych parametów.

Jeśli zależy nam na znajomości macierzy kowariancji, to lepiej posłużyć się do rozwiązania równań metodą Gaussa-Jordana, gdyż daje ona możliwość wyznaczenia odwrotności macierzy $\boldsymbol{A}^T\boldsymbol{A}$. W przeciwnym razie można zastosować rozkład na czynniki trójkątne.

Dokładność parametrów dopasownia

Jakość parametrów dopasowania jest tak długo dobra, jak długo macierz $\boldsymbol{\alpha} = \boldsymbol{A}^T\boldsymbol{A}$ nie jest osobliwa lub prawie osobliwa.

Jeśli

$$\boldsymbol{A} = \begin{bmatrix} 1 & 1 \\ \delta & 0 \\ 0 & \delta \end{bmatrix}$$

i $\delta < \sqrt{\varepsilon}$, to

$$\boldsymbol{A}^T\boldsymbol{A} = \begin{bmatrix} 1+\delta^2 & 1 \\ 1 & 1+\delta^2 \end{bmatrix} \longrightarrow \mathrm{fl}(\boldsymbol{A}^T\boldsymbol{A}) = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$$

Jeśli wskaźnik uwarunkowania $\mathrm{cond}(\boldsymbol{A})$ jest mały, to

$$\mathrm{cond}(\boldsymbol{A}^T\boldsymbol{A}) = \mathrm{cond}(\boldsymbol{A})^2$$

i macierz $\boldsymbol{A}^T\boldsymbol{A}$ jest tym bardziej źle uwarunkowana.

A jako macierz trójkątna górna

Niech $\boldsymbol{A}$ będzie macierzą nadokreśloną ($n > m$). Chcemy znaleźć rozwiązanie takie, że

$$\boldsymbol{Ax} \approx \boldsymbol{b}$$

Jeśli $\boldsymbol{A}$ jest macierzą trójkątną górną, to

$$\begin{bmatrix} \boldsymbol{R} \\ \boldsymbol{O} \end{bmatrix} \boldsymbol{x} \approx \begin{bmatrix} \boldsymbol{b}_1 \\ \boldsymbol{b}_2 \end{bmatrix}$$

$$\|\boldsymbol{r}\|_2^2 = \|\boldsymbol{b} - \boldsymbol{Ax}\|_2^2 = \|\boldsymbol{b}_1 - \boldsymbol{Rx_1}\|_2^2 + \|\boldsymbol{b}_2\|_2^2$$

Szukamy takiego rozwiązania, że

$$\boldsymbol{Rx_1} = \boldsymbol{b}_1$$

które jest dobre w sensie metody najmniejszych kwadratów: minimum równa się $\|\boldsymbol{b}_2\|_2^2$.

Przekształcenia ortogonalne

Jeśli $\boldsymbol{Q}$ jest macierzą ortogonalną, tj. $\boldsymbol{Q}^T\boldsymbol{Q} = \boldsymbol{I}$, to

$$\|\boldsymbol{Q}\boldsymbol{x}\|_2^2 = \boldsymbol{Q}\boldsymbol{x}^T\boldsymbol{Q}\boldsymbol{x} = \boldsymbol{x}^T\boldsymbol{Q}^T\boldsymbol{Q}\boldsymbol{x} = \boldsymbol{x}^T\boldsymbol{x} = \|\boldsymbol{x}\|_2^2$$

Transformacje ortogonalne (obroty, odbicia) przekształcają wektory bez zmiany ich długości.

Można ich użyć do przekształcenia macierzy $\boldsymbol{A}$ w $\boldsymbol{R}$ przy pomocy

- transformacji Givensa (elementarne obroty)
- transformacji Hausholdera (elementarne odbicia)
- ortogonalizacji Grama-Schmidta

Szukamy macierzy $\boldsymbol{Q}$ $(n \times n)$ takiej, że

$$\boldsymbol{A} = \boldsymbol{Q}\begin{bmatrix}\boldsymbol{R}\\ \boldsymbol{O}\end{bmatrix}$$

Przekształcenia ortogonalne – metoda obrotów Givensa

Wektor $\boldsymbol{x} = [x_1 x_2]^T$ jest nachylony pod kątem $\operatorname{tg}\theta = x_2/x_1$ do osi OX.

Po obrocie układu współrzędnych o kąt θ, $\boldsymbol{x} = [\alpha\, 0]^T$.

W ogólności

$$\begin{bmatrix} x_1' \\ x_2' \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} c & s \\ -s & c \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

$$\boldsymbol{x'} = \boldsymbol{Q}\boldsymbol{x}$$

$\boldsymbol{Q}$ jest macierzą ortogonalną

$$\boldsymbol{Q}^T\boldsymbol{Q} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \boldsymbol{I}$$

Metoda obrotów Givensa

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$$

można sprowadzić do postaci trójkątnej górnej traktując pierwszą kolumnę jak współrzędne wektora i dokonując takiego obrotu układu współrzędnych, żeby element a_{21} uległ anihilacji.

Trzeba znaleźć taki kąt θ, żeby

$$\begin{bmatrix} c & s \\ -s & c \end{bmatrix} \begin{bmatrix} a_{11} \\ a_{21} \end{bmatrix} = \begin{bmatrix} \alpha \\ 0 \end{bmatrix}$$

$$-a_{11} \sin\theta + a_{21} \cos\theta = 0$$

- $|a_{11}| > |a_{21}|$: $\operatorname{tg}\theta = a_{21}/a_{11} = t$, $c = \cos\theta = 1/\sqrt{1+t^2}$, $s = \sin\theta = ct$
- $|a_{21}| > |a_{11}|$: $\operatorname{ctg}\theta = a_{11}/a_{21} = \tau$, $s = 1/\sqrt{1+t^2}$, $c = s\tau$

$$\theta \leqslant \pi/4, \qquad \alpha = a_{11}^2 + a_{21}^2$$

Metoda obrotów Givensa

W ogólnym przypadku, dokonujemy eliminacji elementu a_{ti} posługując się obrotem Givensa względem elementu a_{si} $(s < t)$. Macierz tego obrotu jest macierzą obrotu dwuwymiarowego zanurzoną odpowiednio w macierzy jednostkowej $n \times n$.

Np. anihilacja elementu a_{5i} wymaga użycia następującego przekształcenia

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & c & 0 & s & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -s & 0 & c & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} a_{1i} \\ a_{2i} \\ a_{3i} \\ a_{4i} \\ a_{5i} \\ a_{6i} \end{bmatrix} = \begin{bmatrix} a_{1i} \\ a_{2i} \\ a'_{3i} \\ a_{4i} \\ 0 \\ a_{6i} \end{bmatrix}$$

Metoda obrotów Givensa

Anihilacja elementów $(n, 1)$, $(n-1, 1)$, ..., $(2, 1)$, $(n, 2)$, ..., $(3, 2)$, ..., $(n, n-1)$ prowadzi od macierzy $\boldsymbol{A}$ do macierzy trójkątnej górnej.

Jeśli $\boldsymbol{Q} = \boldsymbol{Q_n} \ldots \boldsymbol{Q_2 Q_1}$, to otrzymujemy rozkład macierzy $\boldsymbol{A}$ na macierz ortogonalną i trójkątną górną:

$$\boldsymbol{QA} = \boldsymbol{R} \qquad \boldsymbol{A} = \boldsymbol{Q}^T \boldsymbol{R}$$

Jeśli $\boldsymbol{A}$ jest macierzą układu $\boldsymbol{Ax} = \boldsymbol{b}$, to w wyniku zastosowania obrotów Givensa otrzymujemy równoważny układ $\boldsymbol{QAx} = \boldsymbol{Qb}$.

Jeśli $\boldsymbol{A}$ jest nadokreślona $(n > m)$, to przekształcenia ortogonalne pozwalają wyodrębnić z niej m liniowo niezależych wierszy i kolumn w taki sposób, żeby nie zmienić rozwiązania zagadnienia.

Wynikowy układ równań o macierzy trójkatnej górnej można rozwiązać przez podstawienie wstecz, co prowadzi do uzyskania rozwiązania zadania modelowania danych (zagadnienia aproksymacji średniokwadratowej).

Rozkład wg wartości szczególnych/osobliwych (SVD)

SVD (*Singular Value Decomposition*) to metoda pozwalająca na wyodrębnienie z macierzy prostokątnej $m \times n$ liniowo niezależnych kolumn.

$$\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$$

- $\boldsymbol{U}$ i $\boldsymbol{V}$ są macierzami ortogonalnymi o wymiarach $m \times m$ i $n \times n$
- $\boldsymbol{\Sigma}$ jest macierzą diagonalną $m \times n$ z nieujemnymi elementami rzeczywistymi σ_i[15] uporządkowanymi malejąco, tzn.

$$\sigma_1 \geqslant \sigma_2 \ldots \geqslant \sigma_{\mathsf{min}(m,n)} \geqslant 0$$

σ_i to wartości szczególne $\boldsymbol{A}$, a $\mathsf{min}(m, n)$ kolumn macierzy $\boldsymbol{U}$ i $\boldsymbol{V}$ są lewymi i prawymi wektorami szczególnymi $\boldsymbol{A}$. Mamy

$$\boldsymbol{A}\boldsymbol{v}_j = \sigma_j \boldsymbol{u}_j, \quad \boldsymbol{A}^T\boldsymbol{u}_j = \sigma_j \boldsymbol{v}_j, \quad \boldsymbol{A}^T\boldsymbol{A}\boldsymbol{v}_j = \sigma_j^2 \boldsymbol{v}_j, \quad \boldsymbol{A}\boldsymbol{A}^T\boldsymbol{u}_j = \sigma_j^2 \boldsymbol{u}_j$$

[15] Tutaj σ_i nie mają nic wspólnego z odchyleniem standardowym danych doświadczalnych!

Zastosowania SVD

- Norma euklidesowa

$$\|\boldsymbol{A}\|_2 = \max_{\boldsymbol{x}\neq\boldsymbol{0}} \frac{\|\boldsymbol{A}\boldsymbol{x}\|_2}{\|\boldsymbol{x}\|_2} = \sigma_{\max}$$

- Rząd macierzy. W teorii rząd macierzy jest określony przez liczbę niezerowych wartości szczególnych. W praktyce przez liczbę wartości szczególnych, które są większe od zadanego progu (*rząd numeryczny* macierzy).

Zastosowania SVD

- Wskaźnik uwarunkowania macierzy (prostoktnych i kwadratowych)

$$\mathrm{cond}(\boldsymbol{A}) = \sigma_{\mathsf{max}}/\sigma_{\mathsf{min}}$$

  Dla macierzy kwadratowych ten wskaźnik jest miarą osobliwości, a dla macierzy prostokątnych – liczby liniowo zależnych kolumn.

- Rozwiązywanie równań liniowych dla zagadnień najmniejszych kwadratów. Rozwiązanie minimalizujące normę euklidesową dla równania $\boldsymbol{Ax} \approx \boldsymbol{b}$ wynosi

$$\boldsymbol{x} = \sum_{\sigma_i \neq 0} \frac{\boldsymbol{u}_i^T \boldsymbol{b}}{\sigma_i} \boldsymbol{v}_i$$

  SVD jest szczególnie przydatna przy rozwiązywaniu zagadnień źle uwarunkowanych lub o macierzach o obniżonym rzędzie, gdyż w sumie można opuścić wyrazy ze zbyt małymi wartościami szczególnymi (rozwiązanie jest mniej czułe na zaburzenie danych).

Zastosowania SVD

- Pseudoodwrotność macierzy. Pseudoodwrotnością skalarnej wartości σ jest $1/\sigma$, jeśli $\sigma \neq 0$ lub – w przeciwnym przypadku – zero. To pozwala zdefiniować pseudoodwrotność macierzy diagonalnej Σ^+, która na diagonalnej ma pseudoodwrotności wartości szczególnych. Wówczas

$$\boldsymbol{A}^+ = \boldsymbol{V}\boldsymbol{\Sigma}^+\boldsymbol{U}^T$$

  $\boldsymbol{A}^+$ istnieje zawsze niezależnie od tego, czy macierz jest kwadratowa, czy pełnego rzędu. Jeśli macierz jest kwadratowa i nieosobliwa, to jest ona równoważna macierzy odwrotnej $\boldsymbol{A}^{-1}$.

  Rozwiązanie w sensie minimalizowania normy euklidesowej zagadnienia $\boldsymbol{A}\boldsymbol{x} \approx \boldsymbol{b}$ jest równe $\boldsymbol{A}^+\boldsymbol{b}$.

Zastosowania SVD

- Bazy ortonormalne. Kolumny $\boldsymbol{V}$ odpowiadające zerowym wartościom σ tworzą bazę przestrzeni zerowej $\boldsymbol{A}$ (jądro przekształcenia). Pozostałe wektory $\boldsymbol{V}$ tworzą bazę ortogonalnego uzupełnienia przestrzeni zerowej. Kolumny $\boldsymbol{U}$ odpowiadające niezerowym wartościom szczególnym tworzą bazę przestrzeni będącej obrazem przekształcenia określonego przez macierz $\boldsymbol{A}$. Pozostałe kolumny stanowią bazę ortogonalnego uzupełnienia tej przestrzeni.

Zastosowania SVD

- Przybliżanie macierzy przez macierz o mniejszym rzędzie:

$$\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T = \sigma_1\boldsymbol{E}_1 + \sigma_2\boldsymbol{E}_2 + \ldots + \sigma_n\boldsymbol{E}_n$$

  Każda macierz $\boldsymbol{E}_i = \underline{\mathrm{u}}_i\boldsymbol{v}_i^T$ jest rzędu 1.

  Zwarte przybliżenie macierzy $\boldsymbol{A}$ – suma z pominięciem wyrazów odpowiadających małym wartościom σ_i.

  Jeśli $\boldsymbol{A}$ jest przybliżona przez sumę zbudowaną przy pomocy k największych wartości szczególnych, to otrzymujemy przybliżenie $\boldsymbol{A}$ najlepsze w sensie normy Frobeniusa (norma euklidesowa macierzy traktowanej jako wektor w przestrzeni $\mathcal{R}^{mn}$).

  Takie przybliżenie jest przydatne przy przetwarzaniu obrazów, kompresji danych, kryptografii, itp.

Różnice skończone

Jeśli wartości funkcji $f(x)$ są znane tylko w punktach siatki

$$x_i = x_0 + ih, \qquad i = 0, 1, \ldots, n$$

to znajomość różnych różnic wartości funkcji w tych punktach może być wykorzystana do

- ręcznej i maszynowej interpolacji (śledzenia dokładności tablicowanych wartości)
- numerycznego różniczkowania i całkowania
- numerycznego rozwiązywania zagadnień brzegowych dla równań różniczkowych zwyczajnych
- numerycznego rozwiązywania równań różniczkowych cząstkowych

Progresywne różnice skończone

$$
\begin{aligned}
\Delta^0 f(x) &= f(x) \\
\Delta^1 f(x) &= \Delta[\Delta^0 f(x)] = \Delta^0 f(x+h) - \Delta^0 f(x) = f(x+h) - f(x) \\
\Delta^2 f(x) &= \Delta[\Delta^1 f(x)] = \Delta^1 f(x+h) - \Delta^1 f(x) = f(x+2h) - 2f(x+h) + f(x) \\
&\dots \\
\Delta^k f(x) &= \Delta[\Delta^{k-1} f(x)] = \Delta^{k-1} f(x+h) - \Delta^{k-1} f(x) \\
&= \sum_{i=0}^{k} (-1)^i \binom{k}{i} f(x + (k-i)h), \qquad k = 1, 2, \dots
\end{aligned}
$$

$$
\binom{k}{i} = \frac{k!}{i!(k-i)!}
$$

Wsteczne różnice skończone

$$
\begin{aligned}
\nabla^0 f(x) &= f(x) \\
\nabla^1 f(x) &= \nabla^0 f(x) - \nabla^0 f(x-h) = f(x) - f(x-h) \\
\nabla^2 f(x) &= \nabla^1 f(x) - \nabla^1 f(x-h) = f(x) - 2f(x-h) + f(x-2h) \\
&\dots \\
\nabla^k f(x) &= \nabla[\nabla^{k-1} f(x)] = \nabla^{k-1} f(x) - \nabla^{k-1} f(x-h) \\
&= \sum_{i=0}^{k} (-1)^{k-i} \binom{k}{i} f(x-(k-i)h), \qquad k = 1, 2, \dots
\end{aligned}
$$

Progresywne i wsteczne różnice skończone

$$\begin{aligned}
\nabla f(x) &= f(x) - f(x-h) = \Delta f(x-h) \\
\nabla^2 f(x) &= f(x) - 2f(x-h) + f(x-2h) = \Delta^2 f(x-2h) \\
&\dots \\
\nabla^k f(x) &= \Delta^k f(x-kh)
\end{aligned}$$

Jeśli $x = x_0$, $x_{-i} = x_0 - ih$ i $f_{-i} = f(x_{-i})$, to $\nabla^k f_0 = \Delta^k f_{-k}$.

Ogólnie

$$\nabla^k f_s = \Delta^k f_{s-k}, \qquad \Delta^k f_s = \nabla^k f_{s+k}, \qquad k = 1, 2, \dots$$

Różnice centralne

$$
\begin{aligned}
\delta^0 f(x) &= f(x) \\
\delta^1 f(x) &= f\left(x + \frac{1}{2}h\right) - f\left(x - \frac{1}{2}h\right) \\
\delta^2 f(x) &= f(x+h) - 2f(x) + f(x-h) \\
&\dots \\
\delta^k f(x) &= \delta^{k-1} f\left(x + \frac{1}{2}h\right) - \delta^{k-1} f\left(x - \frac{1}{2}h\right), \qquad k = 1, 2, \dots \\
\delta^{2k} f(x) &= \Delta^{2k} f(x - kh) \\
\delta^{2k+1} f\left(x + \frac{1}{2}h\right) &= \Delta^{2k+1} f(x - kh)
\end{aligned}
$$

Różnice skończone funkcji wielomianowej

Jeśli

$$f(x) = a_0 + a_1 x + a_2 x^2 + \ldots + a_n x^n$$

to

$$\Delta f(x) = f(x+h) - f(x) = \frac{df}{dx} h + \frac{1}{2!} \frac{d^2 f}{dx^2} h^2 + \ldots$$

jest wielomianem stopnia $n - 1$.

Wniosek: Przy ustalonym h n-ta różnica skończona $f(x)$ jest funkcją stałą, a $(n+1)$-a – funkcją stałą równą zeru.

Tablice różnic skończonych

Tablica różnic skończonych dla $f(x) = x^3 - x^2$.

x	$f(x)$	Δ	Δ^2	Δ^3	Δ^4
-2	-12				
		10			
-1	-2		-8		
		2		6	
0	0		-2		0
		0		6	
1	0		4		
		4			
2	4				

Jeśli aproksymujemy funkcję niewielomianową (przestępną, wykładniczą) wielomianem, to tworząc tablicę różnic skończonych możemy obserwować zachowanie się kolejnych różnic i w ten sposób ocenić jakiego stopnia wielomianem (maksymalnie) możemy się w danym przypadku posłużyć.

Interpolacja

Problem: Znamy wartości funkcji $f(x)$ wraz z niektórymi jej pochodnymi na zbiorze pewnych punktów $x_1, x_2, \ldots, x_n$ $(x_1 < x_2 < \ldots < x_n)$, ale nie znamy analitycznego wyrażenia na $f(x)$, która pozwoliłaby na wyznaczenie wartości funkcji (ew. pochodnych) w dowolnym punkcie x.

Rozwiązanie: Przybliżenie funkcji $f(x)$ ciągłą i gładką funkcją przechodzącą przez punkty x_i (zgodność może dotyczyć także pochodnych).

Zadanie interpolacji: wyznaczyć $f(x)$ w punktach nie będących węzłami i oszacować błąd wartości przybliżonych.

- Interpolacja – punkt x leży pomiędzy najmniejszym i największym spośród punktów x_i
- Ekstrapolacja – x leży poza przedziałem wyznaczonym przez x_i

Interpolacja – centralne zagadnienie klasycznych metod numerycznych

Wzór interpolacyjny Lagrange'a

Funkcja $f(x)$ jest określona na węzłach (nierównoodległych) a_j.

Szukamy takich wielomianów $l_j(x)$, żeby funkcja

$$y(x) = \sum_{j=1}^{n} l_j(x) f(a_j)$$

była zgodna z $f(x)$ w węzłach:

$$f(a_j) - y(a_j) = 0, \qquad j = 1, \ldots, n$$

Trzeba wyznaczyć $E(x) = f(x) - y(x)$ w takiej postaci, która pozwoliłaby na wyznaczenie lub oszacowanie błędu przybliżenia.

Z warunku $f(a_j) = y(a_j)$ mamy

$$f(a_k) = \sum_{j=1}^{n} l_j(a_k) f(a_j), \qquad l_j(a_k) = \delta_{jk}, \qquad k = 1, 2, \ldots, n$$

Wzór interpolacyjny Lagrange'a

l_j jest wielomianem takim, że $f_j(a_k) = 0$, $k \neq j$, więc musi zawierać czynnik

$$(x - a_1)(x - a_2)(x - a_{j-1})(x - a_{j+1}) \cdots (x - a_n)$$

Z równości $l_j(a_j) = 1$ wynika, że

$$l_j(x) = \frac{(x - a_1) \cdots (x - a_{j-1})(x - a_{j+1}) \cdots (x - a_n)}{(a_j - a_1) \cdots (a_j - a_{j-1})(a_j - a_{j+1}) \cdots (a_j - a_n)}$$

$l_j(x)$ jest jedynym wielomiane spełniającym warunki $l_j(a_k) = \delta_{jk}$. Nie spełnia ich żaden inny wielomian niższego stopnia.

Można pokazać, że

$$E(x) = \frac{p_n(x)}{n!} f^{(n)}(\xi), \qquad p_n(x) = \prod_{i=1}^{n} (x - a_i)$$

gdzie $\xi \in (a_1, a_n)$. Dla $x = a_i$, $E(a_i) = 0$.

Wzór interpolacyjny Lagrange'a

Wzór

$$f(x) = \sum_{j=1}^{n} l_j(x) f(a_j) + E(x) = y(x) + E(x)$$

definiuje wzór interpolacyjny Lagrange'a.

$n = 2$: wzór interpolacji liniowej

$$y(x) = \frac{x - a_2}{a_1 - a_2} f(a_1) + \frac{x - a_1}{a_2 - a_1} f(a_2)$$

Konstrukcja funkcji $y(x)$ odpowiada znalezieniu funkcji, która przechodzi przez 2 punkty: $(a_1, f(a_1))$ i $(a_2, f(a_2))$.

Ogólnie: konstrukcja $y(x)$ odpowiada znalezieniu funkcji, która przechodzi przez n punktów $(a_j, f(a_j))$, $j = 1, \ldots, n$.

Wszelkie wzory interpolacyjne przyjmujące ustalone wartości we wszystkich węzłach są równoważne (identyczne) ze wzorem interpolacyjnym Lagrange'a.

Ilorazy różnicowe (*divided differences*)

Iloraz różnicowy $f(x)$ dla dwóch argumentów x_k i x_0:

$$[x_k x_0] = \frac{f(x_k) - f(x_0)}{x_k - x_0} = \frac{-(f(x_0) - f(x_k))}{-(x_0 - x_k)} = [x_0 x_k]$$

Dla trzech argumentów x_k, x_0 i x_1:

$$[x_k x_0 x_1] = \frac{[x_k x_0] - [x_0 x_1]}{x_k - x_1} = \frac{[x_k x_0] - [x_1 x_0]}{x_k - x_1}$$

Dla dowolnej liczby argumentów:

$$[x_k x_0 x_1 \ldots x_{n-1}] = \frac{[x_k x_0 x_1 \ldots x_{n-2}] - [x_0 x_1 \ldots x_{n-1}]}{x_k - x_{n-1}}$$

Wzór interpolacyjny Newtona

Ogólny wzór interpolacyjny Newtona otrzymujemy kładąc $x_k = x$

$$
\begin{aligned}
[xx_0x_1\ldots x_n] &= -\frac{[x_0x_1\ldots x_n]}{x-x_n} + \frac{[xx_0x_1\ldots x_{n-1}]}{x-x_n} \\
[xx_0x_1\ldots x_{n-1}] &= -\frac{[x_0x_1\ldots x_{n-1}]}{x-x_{n-1}} + \frac{[xx_0x_1\ldots x_{n-2}]}{x-x_{n-1}} \\
[xx_0x_1\ldots x_{n-2}] &= -\frac{[x_0x_1\ldots x_{n-2}]}{x-x_{n-2}} + \frac{[xx_0x_1\ldots x_{n-3}]}{x-x_{n-2}} \\
[xx_0x_1] &= -\frac{[x_0x_1]}{x-x_1} + \frac{[xx_0]}{x-x_1} \\
[xx_0] &= -\frac{f(x_0)}{x-x_0} + \frac{f(x)}{x-x_0}
\end{aligned}
$$

Wzór interpolacyjny Newtona

Po kolejnych podstawieniach mamy:

$$
\begin{aligned}
[xx_0x_1\ldots x_n] \;=\; & -\frac{[x_0x_1\ldots x_n]}{x-x_n} \\
& -\frac{[x_0x_1\ldots x_{n-1}]}{(x-x_n)(x-x_{n-1})} \\
& -\frac{[x_0x_1\ldots x_{n-2}]}{(x-x_n)(x-x_n)(x-x_{n-1})(x-x_{n-2})} - \ldots \\
& -\frac{[x_0x_1]}{(x-x_{n-1})\ldots(x-x_1)} \\
& -\frac{f(x_0)}{(x-x_n)(x-x_{n-1})\ldots(x-x_1)(x-x_0)} \\
& -\frac{f(x)}{(x-x_n)(x-x_{n-1})\ldots(x-x_1)(x-x_0)}
\end{aligned}
$$

Wzór interpolacyjny Newtona

$$\begin{aligned} f(x) &= f(x_0) + (x - x_0)[x_0x_1] + (x - x_0)(x - x_1)[x_0x_1x_2] + \ldots + \\ &\quad (x - x_0)(x - x_1)\ldots(x - x_{n-1})[x_0x_1x_2\ldots x_n] + R(x) \end{aligned}$$

$$R(x) = (x - x_0)(x - x_1)\ldots(x - x_n)[xx_0x_1x_2\ldots x_n]$$

Jeśli $f(x)$ jest wielomianem, to $R(x)$ jest zerem dla wszystkich x.

Ogólnie $R(x) = 0$ tylko w węzłach.

Pominięcie reszty prowadzi do ogólnego wzoru interpolacyjnego Newtona

$$y(x) = f(x_0) + \sum_{k=0}^{n-1}(x - x_0)(x - x_1)\ldots(x - x_k)[x_0x_1x_2\ldots x_{k+1}]$$

Diagram rombowy/Frasera

$f(x)$	Δ	Δ^2	Δ^3	Δ^4	Δ^5
f_{-2}	$\binom{s+2}{1}$	$\Delta^2 f_{-3}$	$\binom{s+3}{3}$	$\Delta^4 f_{-4}$	$\binom{s+4}{5}$
	Δf_{-2}	$\binom{s+2}{2}$	$\Delta^3 f_{-3}$	$\binom{s+3}{4}$	$\Delta^5 f_{-4}$
f_{-1}	$\binom{s+1}{1}$	$\Delta^2 f_{-2}$	$\binom{s+2}{3}$	$\Delta^4 f_{-3}$	$\binom{s+3}{5}$
	Δf_{-1}	$\binom{s+1}{2}$	$\Delta^3 f_{-2}$	$\binom{s+2}{4}$	$\Delta^5 f_{-3}$
f_0	$\binom{s}{1}$	$\Delta^2 f_{-1}$	$\binom{s+1}{3}$	$\Delta^4 f_{-2}$	$\binom{s+2}{5}$
	Δf_0	$\binom{s}{2}$	$\Delta^3 f_{-1}$	$\binom{s+1}{4}$	$\Delta^5 f_{-2}$
f_1	$\binom{s-1}{1}$	$\Delta^2 f_0$	$\binom{s}{3}$	$\Delta^4 f_{-1}$	$\binom{s+1}{5}$
	Δf_1	$\binom{s-1}{2}$	$\Delta^3 f_0$	$\binom{s}{4}$	$\Delta^5 f_{-1}$
f_2	$\binom{s-2}{1}$	$\Delta^2 f_1$	$\binom{s-1}{3}$	$\Delta^4 f_0$	$\binom{s}{5}$
	Δf_2	$\binom{s-2}{2}$	$\Delta^3 f_1$	$\binom{s-1}{4}$	$\Delta^5 f_0$

$x = x_0 + sh$

Wzory interpolacyjne Newtona-Gregory'ego

- progresywny

$$p_n(x_0 + sh) = f_0 + \binom{s}{1}\Delta^1 f_0 + \binom{s}{2}\Delta^2 f_0 + \ldots + \binom{s}{n}\Delta^n f_0$$

- wsteczny

$$p_n(x_0 + sh) = f_0 + \binom{s}{1}\Delta^1 f_{-1} + \binom{s+1}{2}\Delta^2 f_{-2} + \binom{s+2}{3}\Delta^3 f_{-3} + \ldots + \binom{s+n-1}{n}\Delta^n f_{-n}$$

Wzory interpolacyjne Gaussa

- progresywny

$$p_n(x_0+sh) = f_0 + \binom{s}{1}\Delta^1 f_0 + \binom{s}{2}\Delta^2 f_{-1} + \binom{s+1}{3}\Delta^3 f_{-1} +$$
$$\binom{s+1}{4}\Delta^4 f_{-2} + \binom{s+2}{5}\Delta^5 f_{-2} + \ldots + \binom{s+k-1}{2k}\Delta^{2k} f_{-k} + \binom{s+k}{2k+1}\Delta^{2k+1} f_{-k}$$

- wsteczny

$$p_n(x_0+sh) = f_0 + \binom{s}{1}\Delta^1 f_{-1} + \binom{s+1}{2}\Delta^2 f_{-1} + \binom{s+1}{3}\Delta^3 f_{-2} +$$
$$\binom{s+2}{4}\Delta^4 f_{-2} + \binom{s+2}{5}\Delta^5 f_{-3} + \ldots + \binom{s+k}{2k}\Delta^{2k} f_{-k} + \binom{s+k}{2k+1}\Delta^{2k+1} f_{-k-1}$$

Wzór interpolacyjny Stirlinga

$$
\begin{aligned}
p_{2k+1}(x_0+sh) &= f_0 + \frac{1}{2}\binom{s}{1}\left[\Delta^1 f_0 + \Delta^1 f_{-1}\right] + \\
&\quad \frac{1}{2}\left[\binom{s}{2} + \binom{s+1}{2}\right]\Delta^2 f_{-1} + \frac{1}{2}\binom{s+1}{3}\left[\Delta^3 f_{-1} + \Delta^3 f_{-2}\right] + \\
&\quad \frac{1}{2}\left[\binom{s+1}{4} + \binom{s+2}{4}\right]\Delta^4 f_{-2} + \frac{1}{2}\binom{s+2}{5}\left[\Delta^5 f_{-2} + \Delta^5 f_{-3}\right] + \ldots + \\
&\quad \frac{1}{2}\left[\binom{s+k}{2k+1} + \binom{s+k}{2k}\right]\Delta^{2k} f_{-k} + \frac{1}{2}\binom{s+k}{2k+1}\left[\Delta^{2k+1} f_{-k} + \Delta^{2k+1} f_{-k-1}\right] \\
&= f_0 + \binom{s}{1}\mu\delta f_0 + \frac{s}{2}\binom{s}{1}\delta^2 f_0 + \binom{s+1}{3}\mu\delta^3 f_0 + \ldots + \\
&\quad \frac{s}{2k}\binom{s+k-1}{2k-1}\delta^{2k} f_0 + \binom{s+k}{2k+1}\mu\delta^{2k+1} f_0
\end{aligned}
$$

$$
\mu f(x) = \frac{1}{2}\left[f(x+h/2) + f(x-h/2)\right] \qquad \mu\delta f(x) = \frac{1}{2}\left[\delta f(x+h/2) - \delta f(x-h/2)\right]
$$

Diagram rombowy – wzory interpolacyjne

Wzory interpolacyjne buduje się wg poniższych zasad:

- ścieżka rozpoczyna się zwykle na elemencie f_0 (1. wyraz we wzorze)
- jeśli ścieżka biegnie w prawo w dół, to dodawna jest różnica skończona, na którą trafia, mnożona przez współczynnik dwumienny nad różnicą
- jeśli ścieżka biegnie w prawo w górę, to dodawna jest różnica skończona, na którą trafia, mnożona przez współczynnik dwumienny pod różnicą
- jeśli ścieżka biegnie poziomo i trafia na współczynnik dwumienny, to dodawany wyraz jest średnią arytmetyczną różnic skończonych nad i pod współczynnikiem
- jeśli ścieżka biegnie poziomo i trafia na różnicę skończona, to dodawany wyraz jest średnią arytmetyczną współczynników dwumiennych nad i pod różnicą skończoną
- jeśli poruszamy się w lewo, to wyrazy we wzorze nie są dodawane, ale odejmowane

Każdy wzór interpolacyjny zbudowany tych zasad, który kończy się na n-tej różnicy skończonej jest równoważny pewnemu wzorowi interpolacyjnemu Lagrange'a zbudowanemu w oparciu o węzły użyte do wyznaczenie tej różnicy.[16]

[16] Można pokazać, że dowolna zamknięta droga na diagramie nie wnosi żadnych składników do wzoru interpolacyjnego.

Który wielomian interpolacyjny zastosować?[17].

- Interpolacja w pobliżu początku tablicy (f_0) – wzór progresywny Newtona-Gregory'ego.
- Interpolacja w pobliżu końca tablicy – wzór wsteczny Newtona-Gregory'ego.
- Interpolacja w środku tablicy i dla $s \approx 0$) – wzór Stirlinga.
- Interpolacja w pobliżu środka tablicy i $s > 0$ – wzór progresywny Gaussa.
- Interpolacja w pobliżu środka tablicy i $s < 0$ – wzór wsteczny Gaussa

[17] P. P. Gupta, Sanjay Gupta, G. S. Malik, *Calculus of Finite Differences and Numerical Analysis*

Algorytm Neville'a

Jak interpolować bez wyznaczania wielomianu interpolacyjnego *explicite*?

Mamy zbiór punktów węzłowych $(x_i, f(x_i))$, $n = 0, 1, \ldots, n$. Szukamy wielomianów, które spełniają równości

$$P_{i_0 i_1 \ldots i_k}(x_{i_j}) = f_{i_j}, \quad j = 0, 1, \ldots, k$$

czyli wielomianów przechodzących przez wybrane punkty.

Wielomiany te spełniają następujący związek rekurencyjny

$$\begin{aligned} P_i(x) &\equiv f(x_i) = f_i \\ P_{i_0 i_1 \ldots i_k}(x) &= \frac{(x - x_{i_0}) P_{i_1 \ldots i_k}(x) - (x - x_{i_k}) P_{i_0 i_1 \ldots i_{k-1}}(x)}{x_{i_k} - x_{i_0}} \end{aligned}$$

Wielomiany $P_i(x)$ są wielomianami stopnia zerowego przechodzące przez odpowiadający im punkt x_i.

Algorytm Neville'a

$$P_{01}(x) = \frac{(x - x_0)P_1(x) - (x - x_1)P_0(x)}{x_1 - x_0}$$

$$= \frac{(x - x_0)f_1 - (x - x_1)f_0}{x_1 - x_0} = \frac{x - x_0}{x_0 - x_1}f_0 + \frac{x - x_0}{x_1 - x_0}f_1$$

P_{01} jest wielomianem stopnia pierwszego zgodnym z węzłami (x_0, f_0) i (x_1, f_1).

Analogicznie, $P_{12}(x)$ jest wielomianem stopnia pierwszego zgodnym z węzłami (x_1, f_1) i (x_2, f_2), itd.

Algorytm Neville'a

$$P_{012}(x) = \frac{(x - x_0)P_{12}(x) - (x - x_2)P_{01}(x)}{x_2 - x_0}$$

Łatwo pokazać, że

$$\begin{aligned} P_{012}(x_0) &= f_0 \\ P_{012}(x_1) &= f_1 \\ P_{012}(x_2) &= f_2 \end{aligned}$$

$P_{012}(x)$ jest zgodny z węzłami (x_0, f_0), (x_1, f_1) i (x_2, f_2) i jest to jedyny wielomian drugiego stopnia mający tę własność.

Podobnie, $P_{i_0 i_1 \ldots i_k}(x)$ jest zgodny z węzłami (x_{i_0}, f_{i_0}), (x_{i_1}, f_{i_1}), $\ldots$, (x_{i_k}, f_{i_k}) i jest to jedyny wielomian stopnia k-go o tej własności.

Algorytm Neville'a

W praktyce stosowanie algorytmu Neville'a sprowadza się do utworzenia tabeli

k	0	1	2	3	4
x_0	$f_0 = P_0(x)$				
		$P_{01}(x)$			
x_1	$f_1 = P_1(x)$		$P_{012}(x)$		
		$P_{12}(x)$		$P_{0123}(x)$	
x_2	$f_2 = P_2(x)$		$P_{123}(x)$		$P_{01234}(x)$
		$P_{23}(x)$		$P_{1234}(x)$	
x_3	$f_3 = P_3(x)$		$P_{234}(x)$		
		$P_{34}(x)$			
x_4	$f_4 = P_4(x)$				

Tabelę buduje się tak długo, aż różnice wartości w kolumnie k-tej są mniejsze od żądanej dokładności interpolacji.

Algorytm Neville'a versus algorytm Aitkena

Algorytm Aitkena wykorzystuje wzór rekurencyjny bardzo podobny do stosowanego w algorytmie Neville'a:

$$
\begin{aligned}
P_i(x) &= f_i \\
P_{i_0 i_1 \ldots i_k}(x) &= \frac{(x - x_{i_{k-1}}) P_{i_0 i_1 \ldots i_{k-2} i_{i_k}}(x) - (x - x_{i_k}) P_{i_0 i_1 \ldots i_{k-1}}(x)}{x_{i_k} - x_{i_{k-1}}}
\end{aligned}
$$

Algorytm Aitkena został wyparty przez algorytm Neville'a z uwagi na swoje gorsze własności numeryczne.

Interpolacja funkcjami sklejanymi

Interpolacja funkcjami sklejanymi (*spline*) jest stosunkowo nową metodą, której znaczenie ciągle wzrasta. Jest to dobra metoda do wyznaczania krzywych empirycznych oraz do aproksymacji funkcji skomplikowanych matematycznie.

Niech $\Delta = \{a = x_0 < x_1 < \ldots x_n = b\}$ będzie podziałem odcinka $[a, b]$.

Funkcja sklejana trzeciego stopnia określona na przedziale $[a, b]$

- jest na tym przedziale dwukrotnie różniczkowalna
- na każdym z podpodziałów $[x_i, x_{i+1}]$, $i = 0, \ldots, n-1$ pokrywa się z wielomianem trzeciego stopnia

Funkcja sklejana trzeciego stopnia składa się z wielomianów trzeciego stopnia połączonych razem tak, że ich wartości i wartości ich pierwszych dwóch pochodnych są równe w węzłach x_i, $i = 1, \ldots, n-1$. Problem ma jednoznaczne rozwiązanie przy narzuceniu pewnych dodatkowych warunków na wartości pierwszej/drugiej pochodnej funkcji sklejanej w punktach a i b.

Interpolacja funkcjami wymiernymi

Funkcje wymierne nie dają się dobrze przybliżać przez wielomiany.

Do ich aproksymacji trzeba stosować funkcje postaci

$$R^{\mu\nu}(x) = \frac{P^{\mu}(x)}{Q^{\nu}(x)} = \frac{a_0 + a_1 x + \cdots a_\mu x^\mu}{b_0 + b_1 x + \cdots b_\nu x^\nu} = \frac{a'_0 + a'_1 x + \cdots a'_\mu x^\mu}{1 + b'_1 x + \cdots b'_\nu x^\nu}$$

(μ, ν) – stopień zadania interpolacji wymiernej

Funkcja wymierna $R^{\mu\nu}(x)$ jest określona przez $\mu + \nu + 1$ współczynników $b'_i = b_i/b_0$ i $a'_i = a_i/b_0$. Współczynniki te wyznaczane są z warunków

$$R^{\mu\nu}(x_i) = f_i, \qquad i = 0, 1, \ldots, \mu + \nu$$

Interpolacja funkcjami wymiernymi

Przykład: Należy wyznaczyć wartość funkcji ctg w pobliżu zera.

Bulirsch i Boer zaproponowali algorytm typu Neville'a, który pozwala przeprowadzić interpolację/ekstrapolację funkcjami wymiernymi dla zadanych punktów węzłowych.

Interpolacja wielomianowa i algorytmem B&B w oparciu o wartości funkcji ctg w punktach 1^o, 2^o, ..., 5^o daje:

metoda	$\mathrm{ctg}(2^o30')$
f. wielomianowa (4. st.)	22.638 171 58
f. wymierna (S&B)	22.903 765 52
dokładna	22.903 765 55

Różniczkowanie numeryczne

- Jeśli funkcja jest dana w postaci tabeli swoich wartości, to przed jej zróżniczkowaniem trzeba ją przybliżyć gładką funkcją. Tę funkcję można różniczkować analitycznie lub numerycznie.
- Jeśli $p_n(x)$ jest dobrym wielomianowym przybliżeniem funkcji $f(x)$, to $p'_n(x)$ można używać jako przybliżenia funkcji $f'(x)$.
- Wzory różnicowe do obliczania pierwszej i wyższych pochodnych można otrzymać przez wielokrotne różniczkowanie wzorów interpolacyjnych (Lagrange'a, Newtona, Gaussa, Stirlinga).

Wg wzoru progresywnego Newtona

$$p_n(x) = f(x_i + sh) = f_i + \binom{s}{1}\Delta f_i + \binom{s}{2}\Delta^2 f_i + \ldots + \binom{s}{n}\Delta^n f_i$$
$$= f_i + s\Delta f_i + \frac{s(s-1)}{2}\Delta^2 f_i + \frac{s(s-1)(s-2)}{3!}\Delta^3 f_i + \ldots$$

to

$$\frac{dp_n(x)}{dx} = \frac{dp_n(x_i+sh)}{ds}\frac{ds}{dx} = \frac{1}{h}\frac{dp_n(x_i+sh)}{ds}$$
$$= \frac{1}{h}\left[\Delta f_i + \left(s - \frac{1}{2}\right)\Delta^2 f_i + \left(\frac{3s^2}{6} - \frac{6s}{6} + \frac{2}{6}\right)\Delta^3 f_i\right]$$
$$= \frac{1}{h}\left[\Delta f_i + \left(s - \frac{1}{2}\right)\Delta^2 f_i + \left(\frac{1}{2}s^2 - s + \frac{1}{3}\right)\Delta^3 f_i + \ldots\right]$$

Dla $x = x_i$ $(s = 0)$

$$p_n^{(1)}(x_i) = \frac{1}{h}\left[\Delta f_i - \frac{1}{2}\Delta^2 f_i + \frac{1}{3}\Delta^3 f_i + \ldots\right]$$

Po rozpisaniu różnic skończonych mamy

$$\begin{aligned} p_n^{(1)}(x_i) &= \frac{f_{i+1} - f_i}{h} + O(h) \\ p_n^{(1)}(x_i) &= \frac{-f_{i+2} + 4f_{i+1} - 3f_i}{2h} + O(h^2) \end{aligned}$$

$$\frac{d^2 p_n(x)}{dx^2} = \frac{1}{h^2} \frac{d^2 p_n(x_i + sh)}{ds^2} = \frac{1}{h^2} \left[\Delta^2 f_i + (s-1)\, \Delta^3 f_i + \ldots \right]$$

$$\begin{aligned} p_n^{(2)}(x_i) &= \frac{1}{h^2} \left[\Delta^2 f_i - \Delta^3 f_i + \ldots \right] \\ p_n^{(2)}(x_i) &= \frac{-f_{i+2} + 2f_{i+1} + f_i}{h^2} + O(h) \\ p_n^{(2)}(x_i) &= \frac{-f_{i+3} + 4f_{i+2} - 5f_{i+1} + 2f_i}{h^2} + O(h^2) \end{aligned}$$

Wg wzoru Stirlinga

$$\begin{aligned} p_n^{(1)}(x_i) &= \frac{1}{h}\mu\left(\delta f_i - \frac{1}{6}\delta^3 f_i + \frac{1}{30}\delta^5 f_i - \frac{1}{140}\delta^7 f_i + \ldots\right) \\ &= \frac{f_{i+1} - f_{i-1}}{2h} + O(h^2) \\ &= \frac{-f_{i+2} + 8f_{i+1} - 8f_{i-1} + f_{i-2}}{12h} + O(h^4) \end{aligned}$$

$$\begin{aligned} p_n^{(2)}(x_i) &= \frac{1}{h^2}\left(\delta^2 f_i - \frac{1}{12}\delta^4 f_i + \frac{1}{90}\delta^6 f_i - \frac{1}{560}\delta^8 f_i + \ldots\right) \\ &= \frac{f_{i+1} - 2f_i + f_{i-1}}{h^2} + O(h^2) \\ &= \frac{-f_{i+2} + 16f_{i+1} - 30f_i + 16f_{i-1} - f_{i-2}}{12h^2} + O(h^4) \end{aligned}$$

Dokładność

Wartości funkcji są poprawnie zaokrąglone do r cyfr po kropce.

Błąd zaokrąglenia dla wzoru interpolacyjnego Lagrange'a

$$|E(x)| \leqslant 5 \times 10^{-r-1} \sum_{j=1}^{n} |l_j(x)|$$

Błąd zaokrąglenia dla pochodnych

$$|E(x)| \leqslant 5 \times 10^{-r-1} \frac{1}{h^k} \sum_{j=1}^{n} |l_j^{(k)}(x)|$$

Wzór Lagrange'a:

- interpolacja: błąd metody $\propto h^q$, błąd zaokrąglenia nie zależy od h.
- różniczkowanie: błąd metody $\propto h^p$, błąd zaokrąglenia $\propto 1/h^k$

Wniosek: Przy różniczkowaniu numerycznym należy wybrać h tak, aby błędy metody były porównywalne z błędami zaokrąglenia.

Ekstrapolacja (iterowana) Richardsona

Potrafimy wyznaczyć $F(h)$. Jak uzyskać wynik dla $h \to 0$?

Niech

$$F(h) = a_0 + a_1 h^p + O(h^r), \quad h \to 0, \quad r > p$$

Jeśli potrafimy obliczyć $F(h)$ i $F(qh)$, $(q > 1)$, to

$$\begin{aligned} F(h) &= a_0 + a_1 h^p + O(h^r) \\ F(qh) &= a_0 + a_1 (qh)^p + O(h^r) \\ F(0) = a_0 &= F(h) + \frac{F(h) - F(qh)}{q^p - 1} + O(h^r) \end{aligned}$$

Takie postępowanie nazywa się ekstrapolacją Richardsona.

Jeśli znana jest postać rozwinięcia funkcji $F(h)$ względem potęgi h (nie znamy współczynników rozwinięcia, jedynie jego postać!), to można wówczas wielokrotnie zastosować ekstrapolację Richardsona, co prowadzi do tzw. iterowanej ekstrapolacji Richardsona.

Całkowanie numeryczne

Jeśli funkcja $f(x)$ jest ciągłą funkcją rzeczywistą zmiennej rzeczywistej i jest całkowalna na każdym przedziale $[a, x]$, gdzie $x \in [a, b]$, to

$$F(x) = \int_a^x f(t)dt$$

jest funkcją ciągłą na przedziale $[a, b]$ i różniczkowalną wewnątrz tego przedziału oraz zachodzi równość $f(x) = F'(x)$ (dla punktów wewnątrz przedziału). Ponieważ $F(a) = 0$

$$\int_a^x f(t)dt = F(x) - F(a)$$

Z równoważności całkowania i różniczkowania wynika

$$\frac{dF}{dx} = f(x)$$

Kiedy wyznaczać $F(x)$ przez całkowanie, a kiedy przez równanie różniczkowe?

Kwardratury otwarte i zamknięte

Kwadratury numeryczne służą do wyznaczania całki oznaczonej

$$\int_a^b f(x)dx$$

przez przybliżenie całki sumami skończonymi odpowiadającymi podziałowi przedziału całkowania.

Kwardratury zamknięte korzystają z $f(a)$ i $f(b)$.

Kwadratury otwarte wykorzystują tylko punkty wewnętrzne $[a, b]$.

Kwadratury Newtona-Cotesa

Kwadratury N-C otrzymuje się zastępując funkcję podcałkową wielomianem interpolacyjnym $p_n(x)$:

$$\int_a^b p(x)dx \approx \int_a^b f(x)dx$$

Niech $x_i = a + ih$, $i = 0, 1, \ldots, n$, $h = (b-a)/n$, $n > 0$ i niech $p_n(x)$ będzie wielomianem interpolacyjnym stopnia n lub mniejszego

$$p_n(x_i) = f_i = f(x_i), \quad i = 0, 1, \ldots, n$$

Ze wzoru interpolacyjnego Lagrange'a mamy

$$p_n(x) = \sum_{i=0}^{n} l_i(x) f_i, \quad l_i(x) = \prod_{\substack{k=0 \\ k \neq i}}^{n} \frac{x - x_k}{x_i - x_k}$$

Kwadratury Newtona-Cotesa

Zamiana zmiennych $x = a + ht$ daje

$$l_i(x) = \prod_{\substack{k=0 \\ k \neq i}}^{n} \frac{t-k}{i-k}$$

$$\begin{aligned} \int_a^b p_n(x)dx &= \sum_{i=0}^{n} f_i \int_a^b l_i(x)dx = h \sum_{i=0}^{n} f_i \int_0^n \phi_i(t)dt \\ &= h \sum_{i=0}^{n} f_i \alpha_i \end{aligned}$$

Jest to ogólny wzór na kwadratury Newtona-Cotesa z wagami α_i zależnymi jedynie od n i spełniającymi warunek

$$\sum_{i=0}^{n} \alpha_i = n$$

Kwadratury Newtona-Cotesa

Kwadratura trapezów ($n = 1$, $\alpha_0 = \alpha_1 = 1/2$):

$$\int_a^b p_n(x)dx = \frac{1}{2}h(f_0 + f_1)$$

Kwadratura Simpsona ($n = 2$, $\alpha_0 = 1/3$, $\alpha_1 = 4/3$, $\alpha_2 = 1/3$):

$$\int_a^b p_n(x)dx = \frac{1}{3}h(f_0 + 4f_1 + f_2)$$

n	s	α_i/s	reszta	nazwa	dokładność
1	$\frac{1}{2}$	1 1	$\frac{1}{12}h^3 f^{(2)}(\xi)$	trapezów	x
2	$\frac{1}{3}$	1 4 1	$\frac{1}{90}h^5 f^{(4)}(\xi)$	Simpsona	$\leqslant x^3$
3	$\frac{1}{3}$	1 3 3 1	$\frac{3}{80}h^5 f^{(4)}(\xi)$	Simpsona 3/8	$\leqslant x^3$
4	$\frac{1}{45}$	14 64 24 64 14	$\frac{8}{945}h^7 f^{(6)}(\xi)$	Milne'a-Bode'a	$\leqslant x^5$

Kwadratury złożone

Kwadratur N-C dla $n > 6$ nie stosuje się z uwagi na ich złe własności numeryczne.

Do przybliżania całki w całym przedziale $[a, b]$ stosuje się kwadratury złożone.

Złożona kwadratura trapezów:

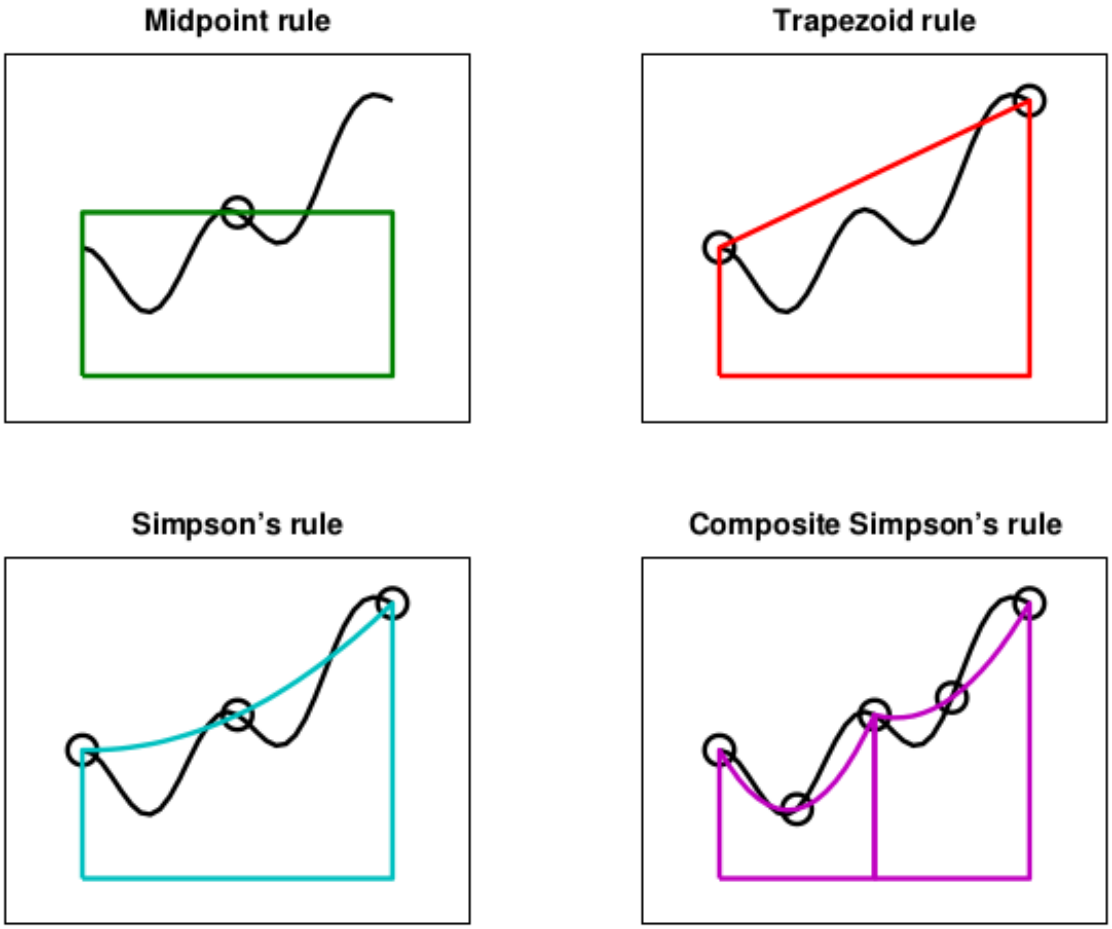
$$\begin{aligned}\int_a^b f(x)dx \approx \sum_{k=0}^{n-1} I_k^T &= \sum_{k=0}^{n-1} \frac{1}{2} h \left(f_k + f_{k+1}\right) \\ &= \frac{1}{2} h \left(f_0 + 2f_1 + 2f_2 + \ldots + 2f_{n-1} + f_n\right) \\ &= h \left(\frac{1}{2}(f_0 + f_n) + (f_1 + f_2 + \ldots + f_{n-1})\right)\end{aligned}$$

Kwadratury złożone

Złożona kwadratura Simpsona ($n = 2m$):

$$\begin{aligned}
\int_a^b f(x)dx &\approx \sum_{k=0}^{n-1} I_k^S \\
&= \sum_{k=0}^{n-1} \frac{1}{3} h \left(f_k + 4f_{k+1} + f_{k+2} \right) \\
&= \frac{1}{3} h \left(f_0 + 4f_1 + 2f_2 + 4f_3 + \ldots + 2f_{n-2} + 4f_{n-1} + f_n \right) \\
&= \frac{1}{3} h \left[(f_0 + f_{2m}) + 2(f_2 + f_4 + \ldots + f_{2(m-1)}) + 4(f_1 + f_3 + \ldots + f_{2m-1}) \right]
\end{aligned}$$

Kwadratury numeryczne: porównanie[18]



[18] http://www.mathworks.com/moler/quad.pdf

Kwadratury otwarte

Otwarte kwadratury N-C nie mają praktycznego zastosowania, bo trudno z nich tworzyć kwadratury złożone. Są one przydatne do uzupełnienia kwadratur złożonych, np.

$$\begin{aligned}
\int_{x_0}^{x_1} f(x)dx &= hf_1 + O(h^2 f^{(1)}) \\
&= \frac{1}{2}h(3f_1 - f_2) + O(h^3 f^{(2)}) \\
&= \frac{1}{12}h(23f_1 - 16f_2 + 5f_3) + O(h^4 f^{(3)}) \\
&= \frac{1}{24}h(55f_1 - 59f_2 + 37f_3 - 9f_4) + O(h^5 f^{(4)})
\end{aligned}$$

Wzory powyższe są dokładne dla funkcji stałej, wielomianu stopnia 1-go, 2-go, itd.

Wzór sumacyjny Eulera-MacLaurina

Niech $T(h)$ będzie sumą określoną przez złożony wzór trapezów:

$$T(h) = \frac{1}{2}h(f_0 + 2f_1 + 2f_2 + \ldots 2f_{n-1} + f_n)$$

Wtedy

$$\begin{aligned} T(h) = \int_a^b f(x)dx \; &+ \; \frac{1}{12}h^2[f^{(1)}(b) - f^{(1)}(a)] - \frac{1}{720}h^4[f^{(3)}(b) - f^{(3)}(a)] \\ &+ \; \frac{1}{30240}h^6[f^{(5)}(b) - f^{(5)}(a)] + \ldots \\ &+ \; c_{2r}h^{2r}[f^{(2r-1)}(b) - f^{(2r-1)}(a)] + O(h^{(2r+1)}) \end{aligned}$$

$a = x_0$, $b = x_n$ i $h = (b-a)/h$, współczynniki c_{2r} wyrażają się przez tzw. liczby Bernoulliego:

$$c_{2r} = \frac{(-1)^{r+1}B_r}{(2r)!}$$

Wzór sumacyjny Eulera-MacLaurina: zastosowania

- teoretyczna podstawa metody całkowania numerycznego, tzw. metody Romberga
- b. dokładne całkowanie funkcji, których pochodne na krańcach przedziału są znane
- dokładność wzoru trapezów, gdy $f^{(k)}(b) = f^{(k)}(a)$, $k = 1, 3, \ldots$; wzór trapezów jest bardzo dokładny dla funkcji okresowych, o okresie równym długości przedziału całkowania

Metoda Romberga

Ze wzoru summacyjnego E-M wynika, że

$$T(h) = c_0 + c_1 h^2 + c_2 h^4 + \ldots \qquad c_0 = \int_a^b f(x)dx$$

- Ekstrapolacja iterowana Richardsona zastosowana do poprawienia wartości całki otrzymanej przy pomocy wzoru trapezów daje metodę Romberga.
- Jednokrotne zastosowanie ekstrapolacji Richardsona dla złożonej kwadraury trapezów jest równoważne zastosowaniu złożonej metody Simpsona.
- Metoda Romberga pozwala na automatyczne dobranie właściwego kroku całkowania.

Kwadratury Gaussa

Kwadratury N-C: węzły mają ustalone położenie, wagi (α_i) są dobierane tak, aby przybliżenie było ścisłe dla wielomianów określonego stopnia.

Kwadratury Gaussa: węzły i wagi ($2n+2$ parametry) są dobiera tak, aby przybliżenie było ścisłe dla wielomianów stopnia $2n+1$ włącznie.

Ogólna postać:

$$\int_a^b f(x)dx = \sum_{i=0}^{n} w_i f(x_i) + K_{2n+2} f^{(2n+2)}(\xi)$$

Odcięte x_i, wagi w_i oraz współczynniki $K_{2n+2} f^{(2n+2)}(\xi)$ są stablicowane (zob. podręczniki analizy numerycznej).

Kwadratury Gaussa

Plusy:

- wzory Gaussa zwykle dają lepsze wyniki (przy tym samym nakładzie pracy), niż odpowiednie wzory oparte na węzłach równoodległych
- dokładne dla całek postaci $\int_a^b w(x)f(x)dx$, gdzie $f(x)$ jest wielomianem

Podstawowe twierdzenie dotyczące kwadratur Gaussa mówi, że odcięte n-punktowej kwadratuy Gaussa z funkcją wagową $w(x)$ w przedziale (a, b) są dokładnie pierwiastkami wielomianu ortogonalnego dla tej samej funkcji wagowej i tego samego przedziału i są dokładne dla wielomianów stopnia $2n - 1$.

Dla $n = 2$ błąd kwadratury Gauss-Legendre'a wynosi $\frac{1}{135}f^{(4)}(\xi)$, a dla $n = 4$ – $\frac{1}{3472875}f^{(8)}(\xi)$.

$w(x)$	przedział	wielomian
1	$(-1, 1)$	Legendre'a
e^{-t}	$(0, \infty)$	Laguerre'a
e^{-t^2}	$(-\infty, \infty)$	Hermite'a

Kwadratury Gaussa

Minusy:

- węzły kwadratur wyższych rzędów nie pokrywają się z węzłami kwadratur niższych rzędów
- trudność szacowania dokładności

$$\int_{-1}^{1} \frac{1}{x^4 + x^2 + 0.9} dx = \begin{cases} 1.585\,026 & \text{3-punktowy} \\ 1.585\,060 & \text{4-punktowy} \\ 1.582\,233 & \text{dokładna wartość} \end{cases}$$

Równania różniczkowe – sformułowanie zagadnienia

Rozważmy równanie różniczkowe pierwszego rzędu

$$\frac{dy}{dx} = f(x, y), \quad y(x_0) = y_0$$

Zakładamy:

- funkcja $f(x, y)$ jest określona i ciągła w pasie $x_0 \leqslant x \leqslant b$, $-\infty < y < \infty$, gdzie x_0 i b są skończone
- istnieje stala L (stała Lipschitza) taka, że dla każdego $x \in [x_0, b]$ i dowolnych liczb y i y^* zachodzi

$$|f(x, y) - f(x, y^*)| \leqslant L|y - y^*|$$

Przy tych założeniach można udowodnić, że w przedziale $[x_0, b]$ istnieje dokładnie jedna funkcja ciągła i różniczkowalna $y(x)$ spełniająca powyższe równanie.

Równania różniczkowe a procesy fizyczne

Jeśli RR ma opisywać jakiś proces fizyczny, to oczekujemy, że niewielka zmiana warunków początkowych tylko nieznacznie wpłynie na rozwiązanie w okolicy warunków początkowych (nie można tego zakładać przy $x \to \infty$).

Zakładamy, że $f(x, y)$ jest ciągła i że RR ma dla każdej wartości początkowej jedyne rozwiązanie

$$y = \phi(x, x_0, y_0)$$

Przy tych założeniach można pokazać, że ϕ jest funkcją cigłą wszystkich swoich argumentów x, x_0, y_0, tzn. dla każdego $\epsilon > 0$ i $X > 0$ istniej takie $\delta > 0$, że gdy $|y_{01} - y_0| < \delta$ i $|x_{01} - x_0| < \delta$, to rozwiązania $y_1(x)$ i $y(x)$ wyznaczone przez warunki brzegowe $y(x_{01}) = y_{01}$ i $y(x_0) = y_0$ spełniają nierówność

$$|y_1(x) - y(x)| < \epsilon \quad \text{dla} \quad -X \leqslant x \leqslant X$$

Algorytmy rozwiązywania równań różniczkowych

1. Całkowanie równania
$$y'(x) = f(x, y)$$
pomiędzy punktami X i $X + \Delta X$ z przedziału $[a, b]$
$$y(X + \Delta X) - y(X) = \int_X^{X+\Delta X} f(x, y(x))dx$$
Najprostszy sposób otrzymywania metod wielokrokowych: f zastępuje się wielomianem interpolacyjnym.

2. Zastąpienie pierwszej pochodnej w równaniu wyrażeniem zawierającym różnice skończone. Np. jeśli
$$y'(X) \approx \frac{1}{\Delta X}(y(X + \Delta X) - y(X))$$
to
$$y(X + \Delta X) \approx y(X) + \Delta X f(X, y(X))$$

Algorytmy rozwiązywania równań różniczkowych

3. Rozwinięcie rozwiązania w pobliżu punktu $y(X + \Delta X)$ przy pomocy wzoru Taylora

$$y(X + \Delta X) \approx y(X) + \Delta X f(X, y(X)) + \frac{1}{2}(\Delta X)^2 y''(X) + \dots$$

   To podejście prowadzi do otrzymania metod jednokrokowych.

Metody różnicowe: wartości funkcji $f(x, y)$ oblicza się jedynie w punktach (x_i, y_i), $y_i = y(x_i)$.

Metody Rungego-Kutty: wartości funkcji $f(x, y)$ oblicza się w punktach różnych od (x_i, y_i).

Wybór metody rozwiązywania równań różniczkowych

Problemy:

- dokładność i stabilność, analiza przenoszenia się błędów (metoda stabilna, jeśli błąd nie ma tendencji do wzrastania)
- wybór warunków początkowych: niektóre metody rozwiązywania równań różniczkowych wymagają więcej niż jednej wartości początkowej; potrzebne są dodatkowe metody rozpoczynania obliczeń
- szybkość metody: względy praktyczne wymuszają ocenę metody pod względem jej szybkość i wygodę operowania wartościami zmiennych

Metody różnicowe

Rozwiązujemy równanie

$$\frac{dY}{dx} = f(x, Y), \qquad Y(x_0) = Y_0$$

gdzie $Y(x)$ oznacza dokładne rozwiązanie.

Niech $Y_i = Y(x_i)$, $Y_i' = \frac{dY(x)}{dx}|_{x=x_i}$, $h = x_{i+1} - x_i$.

Dla przybliżonego rozwiązania

$$y_i = y(x_i), \qquad y_i' = f(x_i, y_i)$$

Funkcja $y(x)$ istnieje tylko w punktach x_i!

Metody różnicowe

Ogólna postać metod różnicowych p-krokowych

$$y_{n+1} = \sum_{i=0}^{p} a_i y_{n-i} + h \sum_{i=-1}^{p} b_i y'_{n-i}$$

x_n oznacza ostanią wartość, dla której obliczono y, a $p+1$ jest liczbą wartości y potrzebnych do obliczenia y_{n+1}.

- a_i i b_i są dowolne, ale $a_p \neq 0$, $b_p \neq 0$
- jeśli $b_{-1} = 0$, to y_{n+1} jest funkcją jedynie wcześniej obliczonych wartości; otrzymujemy bezpośrednie/ekstrapolacyjne wzory różnicowe
- jeśli $b_{-1} \neq 0$, to $y'_{n+1} = f(x_{n+1}, y_{n+1})$ i rów. musi być rozwiązywane metodą iteracyjną; otrzymujemy pośrednie/interpolacyjne wzory różnicowe (bardziej złożone, ale o lepszych własnościach)

Metody różnicowe

W równaniu

$$y_{n+1} = \sum_{i=0}^{p} a_i y_{n-i} + h \sum_{i=-1}^{p} b_i y'_{n-i}$$

dobiera się współczynniki a_i i b_i tak, aby równanie było dokładne dla funkcji $Y(x)$ bedącej wielomianem określonego stopnia r (wzór ma mieć rząd dokładności r).

Niech $y_i = x_i^j$, $j = 0, 1, \ldots, r$. Z powyższego wzoru wynika, że

$$1 = \sum_{i=0}^{p} (-i)^j a_i + j \sum_{i=-1}^{p} (-i)^{j-1} b_i$$

czyli dysponujemy $r + 1$ równaniami na $2p + 3$ nieznanych parametrów a_i i b_i. Jeśli $2p + 3 > r$, to wolne parametry wykorzystuje się do

- zmniejszenia reszty
- zapewnienia najlepszych warunków przenoszenia się błędów
- uzyskanie porządanych własności obliczeniowych (np. zerowania się pewnych współczynników)

Metody różnicowe: przykład

Wyznaczyć współczynniki we wzorze

$$y_{n+1} = a_0 y_n + h(b_{-1} y'_{n+1} + b_0 y'_n)$$

tak, aby miał o rząd dokładności 2. Dla $p = 0$ i $r = 2$ prosty rachunek daje

$$y_{n+1} = y_n + \frac{h}{2}(y'_{n+1} + y'_n)$$

Warto zauważyć, że jeśli

$$\int_{x_n}^{x_{n+1}} g(x)dx = \frac{h}{2}(g_{n+1} + g_n)$$

i $g(x) = y'(x)$, to

$$y_{n+1} = y_n + \frac{h}{2}(y'_{n+1} + y'_n)$$

Metody ekstrapolacyjno-iteracyjne

Dokładność metod rzędu 2:

$$y_{n+1} = y_{n-2} + \frac{3}{2}h(y'_n + y'_{n-1}), \qquad E = \frac{3}{4}h^3 Y^{(3)}(\xi_1)$$

$$y_{n+1} = y_n \quad + \frac{1}{2}h(y'_{n+1} + y'_n), \qquad E = -\frac{1}{12}h^3 Y^{(3)}(\xi_2)$$

Dla wzorów ustalonego rzędu wzory iteracyjne są dokładniejsze od wzorów ekstrapolacyjnych.

Wzory ekstrapolacyje są wykorzystywane jako tzw. wzory wstępne, a wzory interpolacyjne jako wzory korygujące.

Metody ekstrapolacyjno-iteracyjne (*predictor-corrector*).

Metody ekstrapolacyjno-iteracyjne

Metoda ekstrapolacyjno-iteracyjna rzędu 2.:

$$
\begin{aligned}
y_{n+1}^{(0)} &= y_{n-2} + \frac{3}{2}h(y'_n + y'_{n-1}) \\
y'^{(0)}_{n+1} &= f(x_{n+1}, y_{n+1}^{(0)}) \\
y_{n+1}^{(j+1)} &= y_n + \frac{1}{2}h(y'^{(j)}_{n+1} + y'_n), \qquad j = 0, 1, \ldots
\end{aligned}
$$

Metoda Milne'a

Metoda ekstrapolacyjno-iteracyjna rzędu 4.:

$$y_{n+1}^{(0)} = y_{n-3} + \frac{4}{3}h(2y'_n + y'_{n-1} + 2y'_{n-2}), \qquad E = \frac{14}{45}h^5 Y^{(5)}(\xi_1)$$

$$y'^{(0)}_{n+1} = f(x_{n+1}, y_{n+1}^{(0)})$$

$$y_{n+1}^{(j+1)} = y_{n-1} + \frac{1}{3}h(y'^{(j)}_{n+1} + 4y'_n + y'_{n-1}), \quad j = 0, 1, \ldots, \quad E = -\frac{1}{90}h^5 Y^{(5)}(\xi_2)$$

Oszacowanie błędu: E $\approx -\frac{1}{29}(y_{n+1} - y_{n+1}^{(0)})$

Metody Rungego-Kutty

Ogólna postać:

$$y_{n+1} - y_n = \sum_{i=1}^{n} w_i k_i,$$

gdzie w_i są stałymi współczynnikami, a

$$k_i = h_n f(x_n + \alpha_i h_n, y_n + \sum_{j=1}^{i-1} \beta_{ij} k_j)$$

$h_n = x_{n+1} - x_n$, $\alpha_1 = 0$.

Przy danych wartościach w_i, α_i i β_i otrzymujemy metodę samostartującą. Krok h_n może ulegać zmianie na każdym etapie obliczeń.

Metody R-K wymagają wartości $f(x, y)$ w punktach pośrednich, nie tylko (x_n, y_n)!

Metody Rungego-Kutty

Popularna metoda 4. rzędu ($\alpha_2 = \alpha_3 = 1/2, \alpha_4 = 1$):

$$y_{n+1} - y_n = \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$

$$\begin{aligned} k_1 &= h_n f(x_n, y_n) \\ k_2 &= h_n f(x_n + \frac{1}{2}h_n, y_n + \frac{1}{2}k_1) \\ k_3 &= h_n f(x_n + \frac{1}{2}h_n, y_n + \frac{1}{2}k_2) \\ k_4 &= h_n f(x_n + h_n, y_n + k_3) \end{aligned}$$

Metody różnicowe i Rungego-Kutty: porównanie

1. metody R-K są samostartujące
2. dokładność metod R-K jest porównywalna lub lepsza od metod E-I odpowiedniego rzędu; trudniejszy problem szacowania błędu (trzeba ostrożnie dobierać kroku h)
3. metoda R-K wymaga w każdym etapie obliczenia wartości $f(x, y)$ tyle razy, ile wynosi rząd metody; metody E-I wymagają połowy takich operacji

Z uwagi na punkty 2. i 3. należy stosować metody E-I.

Metody R-K są wygodne do wyznaczania wartości wstepnych.